



Universidad Rey Juan Carlos

**Aprendizaje Interactivo de Algoritmos Voraces: del Enfoque
Individual al Colaborativo**

TESIS DOCTORAL

Ouafae Debdi

2014



Universidad Rey Juan Carlos

**Aprendizaje Interactivo de Algoritmos Voraces: del Enfoque
Individual al Colaborativo**

TESIS DOCTORAL

Doctorando:

Ouafae Debdí

Director:

J. Ángel Velázquez Iturbide

Codirector:

Maximiliano Paredes Velasco

Dr. D. Jesús Ángel Velázquez Iturbide, Catedrático de Universidad del Departamento de Informática y Estadística de la Universidad Rey Juan Carlos, como director de la Tesis Doctoral “Aprendizaje Interactivo de Algoritmos Voraces: del enfoque Individual al Colaborativo” realizada por el doctorando Ouafae Debdí

Y Dr. D. Maximiliano Paredes Velasco, Profesor Contratado Doctor del Departamento de Informática y Estadística de la Universidad Rey Juan Carlos como codirector de las Tesis Doctoral “Aprendizaje Interactivo de Algoritmos Voraces: del enfoque Individual al Colaborativo” realizada por el doctorando Ouafae Debdí

HACEN CONSTAR

que esta Tesis Doctoral reúne los requisitos necesarios para su defensa y aprobación.

En Móstoles, a 2 de Junio de 2014

Dr. D. Jesús Ángel Velázquez Iturbide

Dr. D. Maximiliano Paredes Velasco

Agradecimientos

Quiero expresar mi gran gratitud a mis directores de tesis:

Jesús Ángel Velázquez Iturbide por su dedicación, paciencia y sobre todo, por su apoyo incondicional y sus sabios consejos que me han guiado durante estos años.

Maximiliano Paredes Velasco por su dedicación, esfuerzo y confianza que ha depositado en mí durante la elaboración de esta tesis doctoral.

Además, tengo que agradecer enormemente:

A mi marido, por animarme, apoyarme en todo momento en mi camino, soportarme en mis caídas, brindarme su voluntad y por su amor y comprensión infinitas.

A mi hija, por cargarme de energía y alegría, por cambiar mi vida y convertirme a un ser más maduro.

A mis padres por creer en mí y escucharme cuando más lo necesité.

A todos los compañeros del Departamento de Lenguajes y Sistemas Informáticos I por sus opiniones y aportes.

Índice General

CAPÍTULO 1: INTRODUCCIÓN	23
1.2 Objetivos	25
1.3 Aportaciones Principales	26
1.4 Estructura de la Memoria	29
CAPÍTULO 2: ESTADO DE LA CUESTIÓN	31
2.1 Los Algoritmos	31
2.1.1 Definición de Problema y de Algoritmo	31
2.1.2 Propiedades de los Algoritmos	32
2.1.3 Técnicas de Resolución de Problemas de Optimización	34
2.2 Enseñanza de los Algoritmos	37
2.2.1 Experimentación con las Propiedades de Algoritmos	39
2.2.2 Experimentación y el Método Científico	43
2.2.3 Dificultades de los Alumnos	44
2.3 Sistemas Interactivos para el Aprendizaje de Algoritmos	46
2.3.1 Sistemas Correctores	47
2.3.2 Sistemas de Visualización de Algoritmos	51
2.3.3 Sistemas de Visualización de Programas	58
2.4 Sistemas CSCL	59
2.4.1 Teorías Educativas	59
2.4.2 Aprendizaje Colaborativo Soportado por Computador: CSCL	61
2.4.3 Sistemas CSCL para la Enseñanza de Algoritmos/Programación	64
2.5 Evaluación de Usabilidad en Sistemas Educativos	68
2.5.1 Definición de Usabilidad	68
2.5.2 Principales Técnicas de Evaluación de Usabilidad	69
2.5.3 Experiencias de Evaluación de Usabilidad de Sistemas Educativos	77

2.6	Evaluación de Motivación de Sistemas Educativos	80
2.6.1	Motivación y sus Dimensiones	80
2.6.2	Métodos de Evaluación de Motivación	81
2.6.3	Trabajos Relacionados con la Evaluación de Motivación	83
2.7	Estilos de Aprendizaje	85
2.7.1	Modelos de Estilos de Aprendizaje	86
2.7.2	Modelo de Felder-Silverman	88
 CAPÍTULO 3: SISTEMAS GREEDEX Y GREEDEXCOL		 91
3.1	Antecedentes y Trabajos Relacionados	91
3.1.1	Método Experimental	93
3.1.2	Sistemas de Apoyo	95
3.1.3	Análisis de Malentendidos	102
3.1.4	Refinamiento del Método Didáctico	104
3.1.5	Eficacia Educativa	105
3.2	Sistema TuMiST	106
3.3	Sistema GreedEx	109
3.3.1	Estudio Bibliográfico	110
3.3.2	Nuevos Problemas Soportados	113
3.3.3	Evolución de GreedEx	116
3.3.4	Hallazgos Experimentales	118
3.4	Ampliación de GreedEx	122
3.4.1	Experimentación Colaborativa	122
	Sistema GreedExCol	122
3.4.3	Tecnologías de Desarrollo	127
3.4.3	Arquitectura de software	130
 CAPÍTULO 4: EVALUACIONES DE USABILIDAD DE GREEDEX		 133
4.1	Evaluación de Usabilidad de GreedEx	133
4.2	Primera Evaluación de Usabilidad mediante Cuestionarios	134
4.2.1	Valoraciones Numéricas sobre Aspectos Generales	134

4.2.2	Valoraciones numéricas sobre Aspectos Concretos	136
4.2.3	Críticas y Sugerencias Recibidas	137
4.2.4	Conclusiones de la Primera Evaluación de GreedEx	139
4.3	Primer Análisis de las Observaciones	139
4.3.1	Resumen de las Observaciones	140
4.3.2	Conclusiones de las Observaciones	144
4.4	Segunda Evaluación de Usabilidad mediante Cuestionarios	145
4.4.1	Valoraciones Numéricas sobre Aspectos Generales	145
4.4.2	Valoraciones Numéricas sobre Aspectos Concretos	147
4.4.3	Críticas y Sugerencias Recibidas	147
4.4.4	Conclusiones de la Segunda Evaluación de GreedEx	149
4.5	Segundo Análisis de las Observaciones	149
4.5.1	Resumen de las Observaciones	150
4.5.2	Conclusiones de las Observaciones	151
4.6	Evolución de GreedEx	152
4.6.1	Evolución de Aspectos Generales de GreedEx	152
4.6.2	Evolución de Aspectos Concretos de GreedEx	153
4.6.3	Análisis de la Evolución de GreedEx	154
CAPÍTULO 5: EVALUACIÓN DE USABILIDAD Y EFICACIA EDUCATIVA DE GREEDEXCOL		157
5.1	Una Evaluación de Usabilidad de GreedExCol mediante Cuestionarios	157
5.1.1	Valoraciones Numéricas sobre Aspectos Generales	158
5.1.2	Valoraciones Numéricas sobre Aspectos Concretos	159
5.1.3	Críticas y Sugerencias Recibidas	160
5.1.4	Análisis de las Observaciones y los logs	162
5.1.5	Conclusiones de la Evaluación de Usabilidad de GreedExCol	164
5.2	Una Evaluación de Eficacia Educativa de GreedExCol	164
5.2.1	Contexto Educativo	165
5.2.2	Diseño Experimental	165
5.2.3	Protocolo y Tareas	166

5.2.4	Resultados Obtenidos	166
5.2.5	Conclusiones de la Evaluación de Eficiencia Educativa de GreedExCol	169

CAPÍTULO 6: EVALUACIONES DE MOTIVACIÓN DE GREEDEXCOL **171**

6.1	Evaluación de Motivación de GreedExCol	171
6.2	Primera Evaluación de Motivación de GreedExCol	172
6.2.1	Análisis de Motivación Global	173
6.2.2	Análisis de las Cuatro Dimensiones de la Motivación	175
6.2.3	Conclusiones de la Primera Evaluación de Motivación	178
6.3	Segunda Evaluación de Motivación de GreedExCol	179
6.3.1	Resultados del Análisis <i>pre</i> Motivación	180
6.3.2	Resultados del Análisis de <i>Post</i> Motivación	182
6.3.4	Análisis <i>pre-post</i> del grupo de Control	185
6.3.5	Análisis <i>pre-post</i> del grupo Colaborativo	188
6.3.6	Conclusiones de la Segunda Evaluación de Motivación	191
6.4	Correlación entre Usabilidad y Motivación de GreedExCol	191
6.4.1	Resultados de la Correlación entre Motivación y Usabilidad	192
6.4.2	Conclusiones de la Correlación entre Usabilidad y Motivación	197
6.5	Correlación entre Motivación y Eficiencia de GreedExCol	199
6.5.1	Resultados de la Correlación entre Motivación y Eficiencia Educativa	199
6.5.2	Resultados de la Correlación entre Aumento de Motivación y Aumento de Conocimiento	200
6.5.2	Conclusiones de la Correlación entre Eficiencia y Motivación	201

CAPÍTULO 7: ANÁLISIS DE ESTILOS DE APRENDIZAJE EN ALUMNOS DE INFORMÁTICA **203**

7.1	Análisis de Estilos de Aprendizaje	203
7.1.1	Resultados del Análisis de Estilos de Aprendizaje para Todos los Grupos	204
7.1.2	Resultados del Análisis de Estilos de Aprendizaje para el Grupo de Control	208
7.1.3	Resultados del Análisis de Estilos de Aprendizaje para el Grupo Colaborativo	209
7.1.4	Análisis de Estilos según Preferencia	210
7.1.4	Conclusiones del Análisis de Estilos de Aprendizaje	212

7.2	Análisis de Correlación entre Eficiencia Educativa y Estilos de Aprendizaje	212
7.2.1	Correlación entre Estilos de Aprendizaje y Eficiencia Educativa: Grupo Control	213
7.2.2	Correlación entre Estilos de Aprendizaje y Eficiencia Educativa: Grupo Colaborativo	214
7.2.2	Conclusiones del Análisis de Correlación entre Estilos y Eficiencia	215
7.3	Análisis de Correlación entre Motivación y Estilos de Aprendizaje	215
7.3.1	Correlación entre Estilos de Aprendizaje y el Aumento de Motivación: Grupo de Control	216
7.3.2	Correlación entre Estilos de Aprendizaje y Aumento de Motivación: Grupo Colaborativo	218
7.3.3	Correlación entre Estilos de Aprendizaje y Motivación: Todos los grupos	219
7.3.3	Conclusiones del Análisis de Correlación entre Estilos y Motivación	220
7.4	Conclusiones Generales	221
	CAPÍTULO 8: CONCLUSIONES	223
8.1	Resumen de los Resultados	223
8.2	Trabajos futuros	225
	ANEXO 1: ENUNCIADOS DE LAS PRÁCTICAS	227
	ANEXO 2: CUESTIONARIO DE USABILIDAD	235
	ANEXO 3: CUESTIONARIO DE MOTIVACIÓN 2012	239
	ANEXO 4: CUESTIONARIO DE MOTIVACIÓN 2013	241
	ANEXO 5: PRUEBA DE CONOCIMIENTO	243
	ANEXO 6: CUESTIONARIO DE ESTILOS DE APRENDIZAJE	249
	REFERENCIAS	255

Índice de Figuras

Ilustración 1. Ciclo de vida del software	69
Ilustración 2. Interfaz de usuario de AMO	98
Ilustración 3. Interfaz de usuario de SEDA.....	100
Ilustración 4. Interfaz de usuario de GreedEx	101
Ilustración 5. Interfaz de usuario de TuMiST	108
Ilustración 6. Aspecto de GreedExCol para un grupo de dos alumnos	126
Ilustración 7. Aspecto de GreedExCol para aceptar/rechazar propuesta	126
Ilustración 8. Arquitectura de red.....	128
Ilustración 9. Acceso de JDBC a la base de datos.....	129
Ilustración 10. Base de datos de GreedExCol	130
Ilustración 11. Los paquetes de GreedExCol	132
Ilustración 12. Valoración sobre aspectos generales (Primera evaluación)	135
Ilustración 13. Preguntas de respuesta abierta (Primera evaluación)	137
Ilustración 14. Valoración sobre aspectos Generales (Segunda evaluación)	146
Ilustración 15. Preguntas de respuesta abierta (Segunda evaluación).....	148
Ilustración 16. Evolución de los aspectos generales de GreedEx	153
Ilustración 17. Valoración sobre aspectos generales de GreedExCol	159
Ilustración 18. Preguntas de respuesta abierta sobre GreedExCol.....	161
Ilustración 19. Diagramas de caja para los grupos analizados	167
Ilustración 20. Estilos de aprendizaje de los dos grupos	205
Ilustración 21. Porcentaje de alumnos Activos/Reflexivos.....	205
Ilustración 22. Porcentaje de alumnos Sensoriales/Intuitivos	206
Ilustración 23. Porcentaje de alumnos Visuales/Verbales.....	207

Ilustración 24. Porcentaje de alumnos Secuenciales/Globales.....	208
Ilustración 25. Estilos de aprendizaje para el grupo de Control.....	209
Ilustración 26. Estilos de aprendizaje para el grupo colaborativo.....	210

Índice de Tablas

Tabla 1. Publicaciones en congresos y revistas.....	26
Tabla 2. Informes técnicos y aportaciones en el SITIAE.....	28
Tabla 3. Las heurísticas de Reeves.....	72
Tabla 4. Un ejemplo del problema de selección de actividades.....	93
Tabla 5. Resultado de la aplicación de diferentes funciones de selección	94
Tabla 6. Resultados sobre diferentes datos de entrada.....	95
Tabla 7. Agrupación de problemas.....	112
Tabla 8. Posibles soluciones de el problema de maximizar el peso de objetos en una mochila	114
Tabla 9. Posibles soluciones Mochila 0/1	116
Tabla 10. Mejoras de GreedEx.....	117
Tabla 11. Mochila.....	119
Tabla 12. Maximizar el número de objetos.....	119
Tabla 13. Selección de actividades.....	120
Tabla 14. Mochila 0/1	120
Tabla 15. Maximizar el peso de objetos en una mochila	121
Tabla 16. Maximizar el número de objetos en dos mochilas	121
Tabla 17. Resultados numéricos de las preguntas generales (Primera evaluación)	135
Tabla 18. Resultados ordenados de la calidad de sus elementos (Primera evaluación).....	136
Tabla 19. Resultados numéricos de las preguntas generales (Segunda evaluación).....	145
Tabla 20. Resultados numéricos de los aspectos concretos (Segunda evaluación).....	147
Tabla 21. Evolución de los resultados numéricos de las preguntas generales	152
Tabla 22. Evolución de los aspectos concretos de GreedEx	153
Tabla 23. Resultados numéricos de las preguntas generales (GreedExCol)	158

Tabla 24. Resultados ordenados de la calidad de sus elementos (GreedExCol).....	159
Tabla 25. Porcentaje de alumnos que usaron algunos elementos de GreedExCol.....	163
Tabla 26. Alumnos participantes en la evaluación de eficacia educativa	166
Tabla 27. Descriptivos de <i>pretest</i> y <i>posttest</i>	167
Tabla 28. Cohen <i>test</i>	168
Tabla 29. Tamaño de las muestras de la primera evaluación de motivación	173
Tabla 30. Descriptivos de los tres grupos participantes	173
Tabla 31. Descriptivos de las dimensiones de motivación de para todos los grupos.....	175
Tabla 32. Alumnos Participantes en la Motivación	179
Tabla 33. Descriptivos de <i>pretest</i>	180
Tabla 34. Descriptivos de <i>posttest</i>	182
Tabla 35. Valores descriptivos de motivación global (grupo de control)	185
Tabla 36. Valores descriptivos de motivación intrínseca (grupo de control).....	186
Tabla 37. Valores descriptivos de regulación identificada (grupo de control).....	186
Tabla 38. Valores descriptivos de regulación externa (grupo de control).....	187
Tabla 39. Valores descriptivos de desmotivación (grupo de control)	188
Tabla 40. Valores descriptivos de motivación (grupo colaborativo)	188
Tabla 41. Valores descriptivos de motivación intrínseca (grupo colaborativo).....	189
Tabla 42. Valores descriptivos de regulación identificada (grupo colaborativo).....	189
Tabla 43. Valores descriptivos de regulación externa (grupo colaborativo).....	190
Tabla 44. Valores descriptivos de desmotivación (grupo colaborativo)	190
Tabla 45. Correlación de aspectos visuales.....	193
Tabla 46. Correlación de aspectos funcionales	194
Tabla 47. Correlación de otros aspectos.....	194

Tabla 48. Correlación de aspectos visuales.....	195
Tabla 49. Correlación de aspectos funcionales	196
Tabla 50. Correlación de otros aspectos.....	197
Tabla 51. Alumnos participantes comunes en la correlación.....	199
Tabla 52. Estilos de aprendizaje.....	203
Tabla 53. Alumnos participantes en la evaluación de estilos de aprendizaje.....	204
Tabla 54. Distribución de los alumnos según los estilos de aprendizaje	204
Tabla 55. Estilos de aprendizaje para el grupo de control.....	208
Tabla 56. Estilos de aprendizaje para el grupo colaborativo.....	209
Tabla 57. Preferencias	210
Tabla 58. Distribución de preferencias en los dos grupos.....	211
Tabla 59. Alumnos comunes en el análisis de estilos y eficiencia.....	213
Tabla 60. Correlación de <i>pearson</i> para el grupo de control	213
Tabla 61. Correlación de <i>spearman</i>	213
Tabla 62. Correlación de <i>pearson</i> del grupo colaborativo	214
Tabla 63. Correlación de <i>spearman</i> del grupo experimental	214
Tabla 64. Correlación de <i>pearson</i> para todos los grupos	215
Tabla 65. Correlación de <i>spearman</i> para todos los grupos	215
Tabla 66. Alumnos comunes en motivación y estilos.....	216
Tabla 67. Correlación de <i>pearson</i> para el grupo de control	216
Tabla 68. Correlación de <i>spearman</i> para el grupo de control	217
Tabla 69. Correlación de <i>spearman</i>	217
Tabla 70. Correlación de <i>pearson</i> del grupo Colaborativo	218
Tabla 71. Correlación de <i>spearman</i> del grupo Colaborativo	218

Tabla 72. Correlación de <i>spearman</i>	219
Tabla 73. Correlación de <i>pearson</i> de todos los grupos	219
Tabla 74. Correlación de <i>spearman</i> de todos los grupos	219

Resumen

La tesis estudia el uso y evaluación de sistemas software educativos para la enseñanza de algoritmos de optimización. Existen distintas técnicas de diseño de algoritmos que pueden usarse para resolver problemas de optimización, bien de forma exacta (algoritmos voraces, programación dinámica, vuelta atrás, etc.) bien de forma aproximada (algoritmos aproximados, algoritmos probabilísticos, etc.). En nuestro caso nos hemos centrado en los algoritmos voraces.

Como punto de partida se partía de un método experimental que propone al alumno descubrir si existen funciones de selección óptimas para un problema de optimización dado y varios sistemas interactivos que lo soportan. La aportación principal de nuestro trabajo de tesis es profundizar en la parte informática de este método didáctico, mediante el desarrollo y evaluación de dos sistemas interactivos que lo soportan: el sistema GreedEx y su ampliación colaborativa GreedExCol.

Podemos distinguir dos etapas en nuestro trabajo. En una primera fase, de forma iterativa se evaluó la usabilidad del sistema GreedEx. La usabilidad se evaluó mediante cuestionarios y observaciones, eliminando deficiencias y mejorando algunas funcionalidades. Inicialmente sólo soportaba dos problemas (el problema de la mochila y selección de actividades), y se extendió para soportar otras cuatro variantes del problema de la mochila. Como resultado paralelo, hemos profundizado en el uso docente del método experimental, obteniendo pautas para su uso educativo eficaz.

En una segunda fase, se desarrolló y evaluó GreedExCol, ampliación colaborativa de GreedEx. GreedExCol da soporte a la discusión entre los miembros de un pequeño grupo de hasta cuatro alumnos. La metodología de trabajo con GreedExCol es la siguiente. Primero, cada alumno realiza individualmente su labor indagatoria experimental. Se comparten y discuten los resultados obtenidos por cada uno de ellos, de forma que puedan hacer una propuesta de grupo razonada y consensuada de las funciones de selección que consideran óptimas.

Para evaluar nuestro método educativo, se ha optado por medir el grado de usabilidad de la herramienta GreedExCol, determinar el nivel de motivación de los alumnos y evaluar la eficacia educativa de nuestro enfoque. La usabilidad se evaluó mediante cuestionarios y logs, detectándose la necesidad de mejoras significativas en su infraestructura de colaboración. También se detectó una

correlación entre la motivación de los alumnos y la usabilidad de los distintos elementos de GreedExCol. En una segunda evaluación, se detectaron mejoras significativas en la motivación de los alumnos y en su aprendizaje.

En una tercera fase, se realizó un análisis de estilos de aprendizaje en alumnos de Informática junto con otro análisis de correlación, bien con motivación, bien con eficiencia educativa para comprobar si los alumnos que aprenden más o se motivan más corresponden a algún estilo de aprendizaje.

De las limitaciones encontradas es la falta de enfoques que integran dos paradigmas educativos como son CSCL y el método experimental. Además son pocas, por no decir ninguna, las herramientas educativas para el aprendizaje de los algoritmos voraces que han sido evaluadas con diferentes enfoques: eficiencia, usabilidad y motivación de los alumnos con resultados satisfactorios. Éste es precisamente la aportación principal del trabajo de esta tesis doctoral.

En definitiva esta tesis recoge como aportación e innovación más destacada en el ámbito de la investigación de aprendizaje de algoritmos un método educativo que combina dos enfoques: el experimental y el colaborativo, junto con métodos de evaluación del mismo desde la dimensión de la usabilidad, la eficiencia y la motivación de los alumnos hacia las herramientas que se desprenden de dicho método.

Capítulo 1: Introducción

La educación es la forma de alcanzar la excelencia pero no sólo en los mejores alumnos, sino también que debe ayudar a cada alumno a alcanzar su desarrollo intelectual óptimo según Bruner (1977). El avance tecnológico puede ayudar en la mejora de la educación. La combinación de la educación y la tecnología hizo nacer a la Informática Educativa que hace uso del ordenador para apoyar a la educación. En las últimas décadas, se han ido desarrollando una gran cantidad y variedad de aplicaciones de apoyo a la enseñanza. Sin embargo, estos sistemas no siempre son eficaces educativamente.

La disciplina de la informática es el estudio sistemático de los procesos algorítmicos que describen y transforman información: su teoría, análisis, diseño, eficiencia, implementación y aplicación (Denning, et al., 1989). Los algoritmos son uno de los temas esenciales en la enseñanza de la Informática, en particular sus distintas técnicas de diseño, como es el caso de los algoritmos voraces. Además, es una materia variada y de difícil aprendizaje, lo cual sugiere que sea conveniente abordarla desde diversos planteamientos tecnológicos y pedagógicos.

Este trabajo se engloba dentro de los esfuerzos que se realizan internacionalmente para desarrollar software más motivador y eficaz educativamente. Hay diversos enfoques técnicos para afrontar este reto, pudiendo integrar nuestro trabajo en el desarrollo de software para el aprendizaje mediante simulación (que en nuestro contexto denominamos experimentación interactiva) y para el aprendizaje colaborativo, y en menor medida para el aprendizaje basado en problemas y el aprendizaje por descubrimiento.

Revisando el trabajo actual realizado en el aprendizaje de los algoritmos voraces, existe un alto número de animaciones y sistemas de visualización para los algoritmos voraces. El portal **AlgoViz** reúne una serie de animaciones y visualizaciones para técnicas de algoritmos voraces (Anon., 2009). Sin embargo sólo existe un número limitado de herramientas educativas que apoyan el aprendizaje activo de los algoritmos voraces y un número aun más bajo en un enfoque de aprendizaje colaborativo.

Por otra parte, la explicación abstracta de los algoritmos voraces en los libros de texto, presentando el algoritmo como una receta, describiendo primero la técnica seguido por un número de

problemas, da pocas oportunidades al profesor para proponer y diseñar los ejercicios de prácticas y en consecuencia, no soportan adecuadamente los objetivos de aprendizaje. Por estos motivos, se diseñó una herramienta educativa para este tipo de algoritmos, denominada GreedEx y basada en el método experimental. En consecuencia, una primera etapa en este trabajo fue la mejora y la ampliación de este sistema educativo para soportar un número mayor y extensible de problemas de optimización. Las mejoras y nuevas funciones agregadas al sistema GreedEx han sido fruto de evaluaciones de usabilidad realizadas durante el desarrollo de este trabajo donde además se han descubierto deficiencias.

Por otro lado, el uso de herramientas colaborativas para apoyar la enseñanza se ha extendido en los últimos años, asumiendo un papel más activo en el aprendizaje. La incorporación del enfoque colaborativo con la ayuda del ordenador (CSCL) ha mostrado una gran mejora en la calidad del aprendizaje, tales como el desarrollo de habilidades de pensamiento de alto nivel (Gokhale, 1995) y la mejora del rendimiento académico mientras los alumnos interactúan e intercambian sus opiniones (Prince, 2004). Para ello, se desarrolló la extensión colaborativa GreedExCol, que se basa en el paradigma CSCL con una perspectiva socio-constructivista que apoya los debates colaborativos entre los alumnos en un enfoque colaborativo para el aprendizaje de los algoritmos voraces. GreedExCol soporta y proporciona dos espacios: un espacio individual donde cada alumno trabaja de forma individual en las primeras etapas de experimentación y un espacio de colaboración que permite compartir los experimentos llevados a cabo por un grupo de alumnos donde participan en una discusión en grupo para elaborar y entregar su propuesta de funciones de selección óptimas.

Las principales diferencias que existen entre el sistema GreedEx y las diferentes herramientas que apoyan el aprendizaje de los algoritmos voraces son las siguientes. En primer lugar, GreedEx no es un sistema de visualización puro, sino un sistema para la experimentación que hace uso auxiliar de la visualización y animación. En segundo lugar, los sistemas de visualización normalmente ayudan en los niveles inferiores o intermedios de aprendizaje (niveles de comprensión o análisis); sin embargo, GreedEx se dirige también al nivel de evaluación. En tercer lugar, los sistemas de visualización se utilizan para ilustrar algoritmos correctos, incluyendo algoritmos voraces óptimos. Sin embargo, pocos sistemas soportan algoritmos alternativos que permiten al profesor abordar conceptos erróneos a los alumnos. En cuarto lugar, no existen herramientas que combinan el método

experimental y el paradigma CSCL (la extensión colaborativa GreedExCol) para apoyar el aprendizaje de los algoritmos voraces.

Por otro lado y como última etapa en este trabajo de tesis, se hizo un análisis de estilos de aprendizaje que busca dar respuesta a varias preguntas como ¿Cuáles son los estilos de aprendizaje de los alumnos de Informática? ¿Existe una relación entre los estilos de aprendizaje propuestos por Felder-Silverman con el rendimiento académico? ¿Influyen los estilos de aprendizaje en la motivación de los alumnos? estos resultados deben tener una gran importancia sobre las estrategias de enseñanza que se utilizan por parte de los docentes.

1.2 Objetivos

El objetivo principal de nuestro trabajo es la mejora o desarrollo de sistemas software para el activo aprendizaje de los algoritmos voraces. Los objetivos específicos que planteamos al empezar nuestro trabajo de tesis doctoral tienen tres dimensiones:

1. Dimensión pedagógica: la experimentación es un nuevo enfoque para la enseñanza de los algoritmos voraces dado su uso para determinar la optimalidad. GreedEx/GreedExCol es un sistema que permite experimentar con la optimalidad de diferentes funciones de selección. Por eso, en primer lugar, se debe profundizar en el uso docente del método experimental para alcanzar niveles de conocimiento superiores. En segundo lugar, el aprendizaje colaborativo según Dillenbourg (1999) es una situación en la que dos o más personas aprenden o intentan aprender algo juntos, lo que implica que una herramienta colaborativa debe disponer de mecanismos que les permite a un grupo de alumnos a aprender juntos. Nuestro método educativo debe basarse en el aprendizaje colaborativo para explotar las interacciones entre los alumnos y fomentar el trabajo en grupo. Estas interacciones crean una actitud positiva y fomentan el aprendizaje. En este sentido, una extensión colaborativa podría soportar un espacio de contribuciones que ayudara al debate dentro de un grupo de alumnos incorporando herramientas que soporten la comunicación y coordinación entre alumnos. En resumen, GreedExCol se basará en técnicas avanzadas de computación, interacción, experimentación y visualización facilitando así la aridez del aprendizaje de algoritmos voraces.
2. Dimensión informática: el diseño y el desarrollo de herramientas para el aprendizaje de algoritmos voraces para soportar el método pedagógico con alto grado de usabilidad y

eficiencia desde distintas perspectivas tecnológicas, así como usarlos en combinación con métodos didácticos adecuados. Para ello, en primer lugar, se debe seguir completando la herramienta GreedEx para mejorar su calidad educativa. En segundo lugar, ampliar GreedEx de forma colaborativa.

3. Dimensión de evaluación: un requisito imprescindible de toda aplicación educativa es que sea usable y fácil de usar, por tanto, en primer lugar, ambos sistemas (GreedEx/GreedExCol) debe someterse a evaluaciones de usabilidad para garantizar su facilidad de uso e incorporar nuevas funciones, y en consecuencia, soportar mejor nuestro método. En segundo lugar, la eficacia educativa de una herramienta sirve para potenciar la calidad del proceso de la enseñanza y el aprendizaje. En nuestro caso, la extensión colaborativa GreedExCol debe someterse a una evaluación de eficiencia educativa mediante la realización de un experimento para medir el nivel de conocimiento del alumno, y en consecuencia, medir la calidad del proceso educativo resultante. En tercer lugar, la motivación del alumno es un factor importante para facilitar el aprendizaje (Ryan & Deci, 2000). Por tanto, se debe analizar la motivación de los alumnos con respecto a nuestro método educativo. En cuarto lugar, para conocer los estilos de aprendizaje en los alumnos de Informático y comprobar si los alumnos que aprenden más o se motivan más corresponden a algún estilo de aprendizaje en concreto, se debe realizar un análisis de estilos de aprendizaje usando alguno de los modelos influyentes y validadas.

Los objetivos específicos están planteados de forma genérica. Sin embargo, ya hemos comentado que la línea de investigación abordada en esta tesis ha desarrollado algunos esfuerzos previos, por lo que finalmente las evaluaciones realizadas sobre cada sistema han sido distintas.

1.3 Aportaciones Principales

Las siguientes tablas recogen todas las aportaciones científicas y publicaciones realizadas a lo largo de esta tesis doctoral, en orden cronológico. La Tabla 1 recoge las publicaciones en congresos y revistas. La primera columna de estas dos tablas recogen los datos de la publicación, mientras que la segunda columna contiene el capítulo de la tesis con la que está relacionada.

Tabla 1. Publicaciones en congresos y revistas

PUBLICACIÓN	CAPÍTULOS RELACIONADOS
-------------	---------------------------

<i>Ayudante interactivo para los algoritmos de Prim y Kruskal</i> : Ouafae Debdi, Juan David Granada y J. Ángel Velázquez Iturbide. XVI Jornadas de Enseñanza Universitaria de la Informática (JENUI 2010), pp. 469-477, 2010.	Capítulo 3 (Debdi, et al., 2010b)
<i>Experimentación interactiva con algoritmos voraces</i> : Ouafae Debdi y J. Ángel Velázquez Iturbide. XVI Jornadas de Enseñanza Universitaria de la Informática (JENUI 2010), pp. 485-492, 2010.	Capítulo 3 (Debdi & Velázquez-Iturbide, 2010a)
<i>Un asistente extensible para la experimentación interactiva con problemas combinatorios</i> : J. Ángel Velázquez Iturbide, Ouafae Debdi, Daniel Gómez García, Jesús del Fresno Ramírez, Manuel Rubio Sánchez y Maximiliano Paredes Velasco. XI Simposio Nacional de Tecnologías de la Información y las Comunicaciones en la Educación (SINTICE 2010), pp. 63-70, 2010.	Capítulo 3 (Velázquez-Iturbide, et al., 2010)
<i>Experimentation with Optimization Problems in Algorithm Courses</i> : J. Ángel Velázquez Iturbide and Ouafae Debdi. International Conference on Computer as a Tool IEEE 2011(EUROCON), pp. 1-4, 2011.	Capítulo 3 (Velázquez-Iturbide & Debdi, 2011)
<i>Observations as a Method to Evaluate a Computer-Based Approach to Learning Algorithms</i> : Ouafae Debdi, J. Ángel Velázquez-Iturbide, Natalia Esteban-Sánchez and Antonio Pérez-Carrasco. The 12th IEEE International Conference on Advanced Learning Technologies (ICALT' 12), IEEE Computer Society, pp. 433-435, 2012.	Capítulo 4 (Debdi, et al., 2012)
<i>Interactive experimentation with algorithms</i> : J. Ángel Velázquez-Iturbide, Cristóbal Pareja-Flores, Ouafae Debdi and Maximiliano Paredes-Velasco. In Computers in Education – Volumen 2, Sergei Abramovich (ed.), Nova Science Publishers, pp. 47-70, 2012.	Capítulo 3 (Velázquez-Iturbide, et al., 2012)
<i>Experiences in Usability Evaluation of Educational Programming Tools</i> : J. Ángel Velázquez-Iturbide, Antonio Pérez-Carrasco and Ouafae Debdi. En Student Usability in Educational Software and Games: Improving Experiences, Carina González (ed.), IGI Global, pp. 241-260, 2013.	Capítulo 4 (Velázquez-Iturbide, et al., 2013b)
<i>GreedEx: A Visualization Tool for Experimentation and Discovery Learning of Greedy Algorithms</i> : J. Ángel Velázquez-Iturbide, Ouafae Debdi, Natalia Esteban-Sanchez and Celeste Pizarro. IEEE Transactions on Learning Technologies, vol. 6. Nº 2, pp. 130-143, 2013.	Capítulo 3 (Velázquez-Iturbide, et al., 2013a)
<i>GreedExCol: Una Herramienta Educativa Basada en CSCL para el Aprendizaje de Algoritmos Voraces</i> : J. Ángel Velázquez-Iturbide, Maximiliano Paredes Velasco y Ouafae Debdi. XIV Simposio Nacional de Tecnologías de la Información y las Comunicaciones en la Educación (SINTICE 2013). pp. 96-103, 2013.	Capítulo 3 (Velázquez-Iturbide, et al., 2013c)

Las aportaciones de este trabajo se enmarcan dentro del ámbito de la informática educativa del grupo LITE (*Laboratory of Information Technologies in Education*) de la Universidad Rey Juan Carlos. La Tabla 2 contiene los informes técnicos y las aportaciones publicadas en las Actas del

Seminario de Investigación en Tecnologías de la Información Aplicadas a la Educación del grupo de investigación LITE.

Tabla 2. Informes técnicos y aportaciones en el SITIAE

PUBLICACIÓN	CAPÍTULOS RELACIONADOS
<i>Revisión Bibliográfica de Problemas Combinatorios Resolubles por Técnicas Básicas de Diseño de Algoritmos:</i> Ouafae Debdi, J. Ángel Velázquez Iturbide y Maximiliano Paredes Velasco. Serie de Informes Técnicos DLSII-URJC, Número 2010-03. Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.	Capítulo 3 (Debdi, et al., 2010)
<i>Trabajos para la mejora de ayudantes interactivos para la docencia de algoritmos combinatorios:</i> Ouafae Debdi, Actas del IV Seminario de Investigación en Tecnologías de la Información Aplicadas a la Educación (SITIAE 2010), pp. 115-146, 2010.	Capítulo 3 (Debdi & Velázquez-Iturbide, 2010c)
<i>Evaluación de Usabilidad en Sistemas de Uso Educativo:</i> Ouafae Debdi y J. Ángel Velázquez Iturbide. Actas del V Seminario de Investigación en Tecnologías de la Información Aplicadas a la Educación (SITIAE 2011), 2011.	Capítulo 2 (Debdi & Velázquez-Iturbide, 2011a)
<i>Una Evaluación de Usabilidad de GreedEx:</i> Ouafae Debdi y J. Ángel Velázquez Iturbide. Serie de Informes Técnicos DLSII-URJC, Número 2011-01. Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.	Capítulo 4 (Debdi & Velázquez-Iturbide, 2011b)
<i>Una Segunda Evaluación de Usabilidad de GreedEx:</i> Ouafae Debdi y J. Ángel Velázquez Iturbide. Serie de Informes Técnicos DLSII-URJC, Número 2012-01. Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.	Capítulo 4 (Debdi & Velázquez, 2012b)
<i>Una Evaluación de Usabilidad de GreedExCol:</i> Ouafae Debdi, J. Ángel Velázquez Iturbide y Maximiliano Paredes Velasco. Serie de Informes Técnicos DLSII-URJC, Número 2012-05. Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.	Capítulo 5 (Debdi, et al., 2012c)
<i>Una Evaluación de Eficacia Educativa de GreedExCol:</i> Ouafae Debdi, Maximiliano Paredes Velasco y J. Ángel Velázquez Iturbide. Serie de Informes Técnicos DLSII-URJC, Número 2013-02. Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.	Capítulo 6 (Debdi, et al., 2013b)
<i>Una Evaluación de Motivación de GreedExCol:</i> Ouafae Debdi, J. Ángel Velázquez y Maximiliano Paredes Velasco Iturbide. Serie de Informes Técnicos DLSII-URJC, Número 2013-03. Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.	Capítulo 6 (Debdi, et al., 2013f)
<i>Análisis de Correlación entre Motivación y Usabilidad con GreedExCol:</i> Ouafae Debdi, Maximiliano Paredes Velasco y J. Ángel Velázquez Iturbide. Un	Capítulo 6 (Debdi, et al., 2013c)

Serie de Informes Técnicos DLSII-URJC, Número 2013-04. Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.	
<i>Una Segunda Evaluación de Motivación de GreedExCol:</i> Ouafae Debdi, Maximiliano Paredes Velasco y J. Ángel Velázquez Iturbide. Serie de Informes Técnicos DLSII-URJC, Número 2013-05. Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.	Capítulo 6 (Debdi, et al., 2013e)
<i>Un Análisis de Correlación entre Motivación y Eficiencia Educativa con GreedExCol:</i> Ouafae Debdi, Maximiliano Paredes Velasco y J. Ángel Velázquez Iturbide. Serie de Informes Técnicos DLSII-URJC, Número 2013-06. Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.	Capítulo 6 (Debdi, et al., 2013a)
<i>Un Análisis de Estilos de Aprendizaje en Alumnos de Informática:</i> Ouafae Debdi, Maximiliano Paredes Velasco y J. Ángel Velázquez Iturbide. Serie de Informes Técnicos DLSII-URJC, Número 2014-05. Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.	Capítulo 7 (Debdi, et al., 2014a)
<i>Análisis de Correlación entre Eficiencia Educativa y Estilos de Aprendizaje:</i> Ouafae Debdi, Maximiliano Paredes Velasco y J. Ángel Velázquez Iturbide. Serie de Informes Técnicos DLSII-URJC, Número 2014-06. Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.	Capítulo 7 (Debdi, et al., 2014b)
<i>Análisis de Correlación entre Motivación y Estilos de Aprendizaje:</i> Ouafae Debdi, Maximiliano Paredes Velasco y J. Ángel Velázquez Iturbide. Serie de Informes Técnicos DLSII-URJC, Número 2014-07. Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.	Capítulo 7 (Debdi, et al., 2014c)

1.4 Estructura de la Memoria

La estructura de la memoria de este trabajo de tesis doctoral es la siguiente:

El Capítulo 2 presenta el estado de la cuestión de diferentes temas. En primer lugar, se presentan las técnicas de diseño de algoritmos de optimización. En segundo lugar, se detalla el estado de la cuestión de la enseñanza de los algoritmos. En tercer lugar, se hace un repaso de los sistemas interactivos para el aprendizaje de algoritmos. En cuarto lugar, se definen los principios generales del enfoque CSCL y se recogen algunos de los sistemas utilizados en la enseñanza de algoritmos/programación. En quinto lugar, se recogen los métodos de evaluación de usabilidad en sistemas educativos incluyendo algunos trabajos realizados en la enseñanza de la algoritmos/programación y finalmente se exploran los métodos de evaluación de motivación en sistemas educativos y se recogen algunas experiencias realizadas. El Capítulo 3 se centra en el método didáctico, los sistemas antecedentes (AMO, SEDA y TuMiST) y los sistemas que soportan

nuestro nuevo enfoque: GreedEx y su extensión colaborativa “GreedExCol”. El Capítulo 4 presenta todas las evaluaciones de usabilidad de GreedEx describiendo las evaluaciones y sus resultados. Además se detalla la evolución que experimentó GreedEx. En el Capítulo 5 se detallan la evaluación de usabilidad y eficacia educativa de GreedExCol. Asimismo el Capítulo 6 se adentra en las evaluaciones de motivación de GreedExCol y el análisis de correlaciones entre motivación bien con eficiencia de aprendizaje, bien con usabilidad. El capítulo 7 se centra en los estilos de aprendizaje en alumnos de informática. Por último, el Capítulo 8 se enfoca en las conclusiones generales del trabajo de la tesis doctoral y líneas futuras de trabajo.

Capítulo 2: Estado de la cuestión

En este capítulo, se explora el estado de la cuestión de varios temas que hemos estado investigando durante el desarrollo de nuestro trabajo de tesis. En primer lugar, se define el concepto de algoritmo y sus propiedades, enunciando las técnicas de diseño de algoritmos de optimización como los algoritmos voraces, algoritmos aproximados y otras técnicas de diseño de algoritmos. En segundo lugar, se detalla el estado de la cuestión de la enseñanza de los algoritmos. En tercer lugar, se enumeran los sistemas interactivos para el aprendizaje de algoritmos. En cuarto lugar, se centra en los principios generales del enfoque CSCL y los sistemas basados en este paradigma utilizados en la enseñanza de algoritmos y programación. En quinto lugar, se exponen los métodos de evaluación de usabilidad en sistemas educativos citando algunos trabajos realizados en la enseñanza de la programación. En último lugar, nos enfocamos en la evaluación de motivación de sistemas educativos, enumerando algunos cuestionarios usados para su evaluación y algunos trabajos relacionados.

2.1 Los Algoritmos

En esta sección se define el concepto de algoritmo y se enumeran sus propiedades, centrándose en las propiedades de corrección, eficiencia y optimalidad.

2.1.1 Definición de Problema y de Algoritmo

La noción de un algoritmo es un concepto básico y esencial que se imparte en los planes de estudios en informática. Un algoritmo es un procedimiento computacional bien definido que toma un valor o un conjunto de valores como entrada y produce otro valor, o conjunto de valores, como salida. Un algoritmo es por lo tanto una secuencia de pasos de cálculo que transforman la entrada a la salida.

El objetivo de toda especificación es expresar de forma precisa y correcta lo que debe hacer un algoritmo y bajo qué condiciones se puede ejecutar. La precondición especifica las restricciones que tienen que cumplir los parámetros de entrada y la post-condición especifica la relación que deben

cumplir los parámetros de entrada y de salida para que el algoritmo produzca el resultado deseado. Un algoritmo es una solución de un problema algorítmico que debe cumplir correctamente las pre-condiciones/post-condiciones (relación E / S) especificados en el problema (Haberman, et al., 2005). En otras palabras, un algoritmo describe un procedimiento de cálculo específico para lograr la relación de entrada/salida (Cormen, et al., 2001). Un algoritmo se puede ver como una herramienta para la resolución de un problema computacional bien especificado. Skeina también (2002) lo define como “el procedimiento de resolución de problemas adecuado que puede resolver todos los casos de un problema”.

2.1.2 Propiedades de los Algoritmos

Un algoritmo puede tener muchas propiedades, algunas son fundamentales como son corrección, eficiencia y optimalidad (en el caso de algoritmos de optimización) y otros menos importantes como son claridad, simplicidad, portabilidad, mantenibilidad, etc. Al diseñar un algoritmo, éste debe ser analizado al menos desde los siguientes puntos de vista:

- 1) Corrección: un algoritmo es correcto si para cualquier entrada válida produce el resultado requerido por la especificación del algoritmo.
- 2) Eficiencia: que se refiere a la cantidad de recursos utilizados por el mismo. Para mejorar el rendimiento debemos buscar el mejor algoritmo (que use la menor cantidad de memoria y tiempo) haciendo comparaciones entre diferentes algoritmos que resuelven el mismo problema.
- 3) Optimalidad: esta propiedad tiene sentido en algoritmos que resuelven problemas de optimización. La solución no sólo debe ser válida sino también óptima.

Cada una de estas propiedades se puede analizar de dos maneras complementarias: formal y experimental. La principal ventaja del análisis formal es que se trata de garantizar la exactitud del algoritmo. Su principal desventaja es la dificultad de la demostración, sobre todo para los algoritmos complejos. Por otro lado, la principal ventaja del análisis experimental es su simplicidad mientras que la principal desventaja es la dificultad de sacar una extrapolación útil y rigurosa.

En primer lugar, la corrección de un algoritmo se afirma cuando se dice que el algoritmo es correcto con respecto a una especificación. Se puede realizar un análisis formal o experimental para comprobar la corrección de un algoritmo:

- 1) Análisis Formal. El objetivo del análisis formal es demostrar que el algoritmo funciona correctamente para cualquier instancia de entrada de datos a través de técnicas de comprobación formales. Por ejemplo Si $\{P\} C \{Q\}$ es un código con la precondición $\{P\}$ y la post-condición $\{Q\}$, entonces $\{P\} C \{Q\}$ es correcto si cada estado inicial posible que satisfaga $\{P\}$ da como resultado un estado final que satisface $\{Q\}$. La idea consiste en usar las reglas formales de la lógica para demostrar que un algoritmo cumple con su especificación.
- 2) Análisis Experimental (Pruebas). Las pruebas constituyen un componente esencial en el diseño y análisis de algoritmos. Se prueba el algoritmo para diferentes casos del problema (datos de entrada diferentes).

En segundo lugar, la propiedad de eficiencia de un algoritmo describe en términos abstractos o "notaciones asintóticas" el rendimiento de un algoritmo. Los principales recursos para ejecutar un algoritmo son: tiempo y espacio (memoria utilizada). La eficiencia también puede analizarse de las dos formas descritas:

- 1) Análisis Formal (Análisis de Complejidad). Se expresa en término de categorías abstractas llamadas órdenes de complejidad. Se utiliza para describir el consumo de recursos, principalmente el tiempo, del algoritmo analizado que ofrece estimaciones del comportamiento de un algoritmo de forma independiente del ordenador en donde está implementado y sin necesidad de ejecutarlo. Para describir analíticamente la eficiencia de un algoritmo se usa la función $t(n)$ (n será la variable que denota el tamaño de los datos y $t(n)$ el tiempo empleado en procesar este algoritmo). para representar la complejidad de un algoritmo como una función del tamaño de la entrada n se utilizan varias notaciones, siendo la más utilizada y conocida la notación O grande.
- 2) Análisis Experimental. Sirve para representar medidas reales del comportamiento de un algoritmo. Se calcula el tiempo de ejecución por segundos o milisegundos de un algoritmo para unos datos de entrada dados.

Por último, la propiedad de optimalidad de un algoritmo es específica para los problemas de optimización. También tenemos dos enfoques:

1) Análisis Formal. Se verifica la optimalidad de un algoritmo usando las reglas formales de la lógica para demostrar que una solución calculada por un algoritmo óptimo es tan buena o mejor que cualquier otra solución factible (que cumple con la especificación del algoritmo).

2) Análisis Experimental. La experimentación puede ser utilizada para detectar la no-optimalidad de un algoritmo (encontrar un contraejemplo).

2.1.3 Técnicas de Resolución de Problemas de Optimización

Los algoritmos de optimización son algoritmos que tratan de encontrar una solución que maximiza o minimiza alguna función. En este diseño de algoritmos, para un problema de optimización, hay que demostrar que el algoritmo siempre ofrece la mejor solución posible (Skiena, 2002).

Existen técnicas de diseño de algoritmos para resolver problemas de optimización. En la programación dinámica, un problema se divide en pequeños sub-problemas para después combinarlos resolviendo problemas mayores. En este enfoque, primero se resuelven los casos pequeños, luego se guardan los resultados y más adelante, cada vez que se necesita un resultado se busca una tabla creada para tal fin (Neapolitan & Naimipour, 1999).

El algoritmo de vuelta atrás (*Backtracking*) puede ser descrito como una búsqueda organizada en un espacio de estados en forma de árbol. El espacio de estados, llamado árbol de búsqueda, se recorre en profundidad, realizando comprobaciones incrementales de validez en cada nodo, intentando podar las ramas que sólo van a conducir a soluciones inválidas. En general, es adecuada para la resolución de problemas combinatorios en los que un número potencialmente grande, pero finito de soluciones, tiene que ser inspeccionado (Alsuwaiyel, 1999).

El algoritmo de ramificación y poda (*branch and bound*) es muy similar a la vuelta atrás salvo que el algoritmo de ramificación y poda no nos limita a ninguna forma particular de recorrer el árbol de búsqueda, que se utilizan cotas y que solamente puede aplicarse a problemas de optimización (Neapolitan & Naimipour, 1999)

En esta sección se enuncian con más detalle las técnicas de diseño conocidas como algoritmos voraces y algoritmos aproximados dado que son el objeto del método experimental mencionado anteriormente.

2.1.3.1 Algoritmos Voraces

Un algoritmo voraz es un algoritmo "miope" que siempre toma la decisión que le parezca más prometedora en ese momento, de tal forma que se coge elementos con un orden determinado donde cada vez tomamos la que se considera "mejor", de acuerdo con algún criterio, sin tener en cuenta las elecciones que se hayan hecho anteriormente o que se harán en el futuro (Neapolitan & Naimipour, 1999). Producen una solución óptima y conducen a soluciones muy eficientes, directas y simples, por eso se utilizan para resolver problemas de optimización. Sin embargo, muchos problemas no se pueden resolver correctamente mediante este algoritmo porque la solución global no es siempre óptima. En este caso, los algoritmos voraces aún pueden utilizarse para obtener soluciones aproximadas (quizá subóptimas), hablándose entonces de heurísticas.

Un algoritmo voraz comienza con un conjunto vacío y se irán añadiendo elementos al conjunto en secuencia hasta que el conjunto representa una solución. Los algoritmos voraces constan de cuatro funciones.

- Función que comprueba si el conjunto seleccionado de candidatos proporciona una solución válida.
- Función de factibilidad que compruebe la viabilidad de un conjunto.
- Función de selección que indica cuál de los candidatos es el más prometedor.
- Función objetivo que queremos maximizar o minimizar, que asocia un valor a cada solución.

El algoritmo se puede formalizar de la siguiente forma (Brassard & Bratley, 1996):

```

Función voraz (C: conjunto): conjunto
  {C es el conjunto de candidatos}
  S ← ∅ {construimos la solución en el conjunto S}
  Mientras C ≠ ∅ y no solución (S) hacer
    x ← seleccionar (C)
    C ← C \ {x}
  Fin mientras
  Si factible (S ∪ {x}) entonces S ← S ∪ {x}

```

Si solución (S) entonces devolver S
Sino devolver «no hay soluciones»

Para probar que un problema de optimización puede ser resuelto mediante un algoritmo voraz se debe demostrar que el problema cumple las dos siguientes propiedades:

- Propiedad de la subestructura óptima: una solución global óptima contiene las soluciones óptimas de todos sus sub-problemas (Cormen, et al., 2001). Es decir, la solución óptima de un problema dado se puede obtener mediante el uso de soluciones óptimas de sus sub-problemas. Una solución óptima global está construida a partir de soluciones óptimas locales.
- Propiedad de la elección voraz: una solución óptima global se puede obtener seleccionando con avaricia una elección localmente óptima. La elección voraz es siempre parte de cierta solución óptima. La elección de la solución se basa en seleccionar la mejor solución que parece mejor en este momento y luego resolver los sub-problemas que se presentan más adelante. La elección hecha por un algoritmo voraz puede depender de decisiones tomadas hasta el momento, pero no en las elecciones futuras o todas las soluciones del sub-problema, dado que un algoritmo voraz nunca reconsidera sus opciones.

Algunos ejemplos de problemas de optimización resolubles de forma óptima mediante algoritmos voraces son el problema del árbol de recubrimiento de coste mínimo (algoritmos de Prim y Kruskal), el problema de los caminos más cortos desde un solo origen (algoritmo de Dijkstra), el problema del cambio de monedas, el problema de la mochila y los códigos de Huffman.

2.1.3.2 Algoritmos Aproximados

Una heurística es un algoritmo voraz que siempre proporciona una solución válida para un problema, pero no siempre es una solución óptima (Brassard & Bratley, 1996). Una heurística se llama algoritmo de aproximación cuando garantiza que la solución a un caso dado no difiere de la solución exacta en más de una cantidad absoluta o relativa.

Algunos ejemplos de problemas para los que se han encontrado buenos algoritmos de aproximación son el problema de la cobertura de vértices y el problema del viajante (*The traveling salesman problem*) (Skiena, 2002)(Cormen, et al., 2001).

2.2 Enseñanza de los Algoritmos

En la enseñanza de algoritmos, los aprendices demuestran obstáculos cognitivos en relación con la noción de un algoritmo (Haberman, et al., 2005). Para apoyar a los alumnos en su propia construcción del conocimiento sobre el concepto de un algoritmo, Perrenet y sus colegas (2005) propusieron cuatro niveles de abstracción para el concepto de algoritmo para saber acerca de los procesos de pensamiento de los alumnos y averiguar el incremento de su nivel de abstracción. El resultado fue el aumento de la media de ese nivel entre los grupos de años sucesivos, así como en los grupos durante el año:

1. Nivel de ejecución: el algoritmo es una ejecución específica en una máquina específica; su tiempo de ejecución está determinado por la máquina.
2. Nivel de programa: el algoritmo es un proceso, descrito por un lenguaje específico de programación ejecutable, el tiempo de ejecución depende de la entrada.
3. Nivel de objeto: el algoritmo no está conectado con un lenguaje de programación específico, sino que puede ser visto como un objeto (en lugar de proceso); mientras que la construcción de un algoritmo, la estructura de datos y las propiedades de invariancia se utilizan; meta propiedades, tales como terminación y "patrones" (módulos algorítmicos) son relevantes, el tiempo de ejecución se considera en términos de magnitud de orden en función de la entrada.
4. Nivel de problema: el algoritmo se puede ver como un cuadro negro; la perspectiva de pensamiento es dado un problema, ¿qué tipo de algoritmo es adecuado?; los problemas se pueden clasificar en algoritmos adecuados; un problema tiene una complejidad intrínseca.

Levitin (2000) identificó varios problemas en la enseñanza del diseño y análisis de algoritmos y sugirió maneras de subsanar las deficiencias detectadas. De estos problemas, por ejemplo que simplemente se expone a los alumnos un algoritmo bien conocido o por agrupación de algoritmos a lo largo de algunas técnicas de diseño generales cuya clasificación tradicional tiene deficiencias dado su incapacidad para clasificar muchos algoritmos importantes. Es más, el proceso de resolución de

problemas algorítmicos generalmente no se discute en los libros de texto de algoritmos. En particular, la parte de la implementación del proceso es casi siempre ignorada. Según Ginat (2007), los libros de texto a menudo se centran en la presentación detallada de las características y construcción del lenguaje y ofrecen un examen bastante limitado de las diversas facetas del proceso de diseño del programa. Velázquez (Velázquez-Iturbide, Aceptado) también señaló las dificultades de aprendizaje de la organización de material sobre algoritmos voraces en los libros de texto:

- Diseño de una función de selección voraz óptima. Es fácil para algunos algoritmos simples (por ejemplo, el problema de cambio de moneda) pero es mucho más exigente para otros algoritmos.
- Implementar un algoritmo voraz. La mayoría de los libros de texto ofrecen una plantilla de alto nivel. Los algoritmos voraces tienen la misma semántica que de la plantilla, pero su código es único para cada algoritmo. Además, algunos algoritmos ni siquiera se ajustan a esta plantilla de alto nivel.
- Demostrar la optimalidad de una función de selección voraz. La mayoría de las demostraciones son por contradicción o al absurdo, sin embargo los alumnos no están muy familiarizados con estas técnicas.

Existen trabajos que han hecho uso de ejercicios o tipos de problemas para ilustrar mejor las técnicas de diseño. Por ejemplo, Walker (1997) proporcionó un ejercicio para motivar la utilización de técnicas de diseño de algoritmos, la estructura del programa y las pruebas. Levitin (2012) subrayó el papel de polinomios “embaldosado con poliominós” que puede proporcionar valiosos ejemplos de aplicaciones de las técnicas de diseño de algoritmos casi totalmente olvidados por los libros de texto sobre algoritmos. Levitin en otra experiencia (2002) enumeró varios problemas tipo rompecabezas (*puzzle*) que pueden ser utilizados para ilustrar la mayoría de las técnicas generales de diseño de algoritmo.

En otras experiencias se integró la asignatura de computadores, introducción a la informática, con asignaturas muy diferentes como es la biología. Según Dodds y sus colegas (2012), la fusión de biología e informática proporciona un contexto rico y atractivo en el que se enseña la informática, junto con potentes herramientas de cálculo con el que explorar los fenómenos biológicos. Esta experiencia muestra que un curso con sólo la mitad del número habitual de horas de clase puede tener éxito. Otro trabajo descrito por D'Antonio(2003) también combinó la bio-informática que estudia

secuencias biológicas como el ADN y Algoritmos que involucran la búsqueda y la combinación de tales secuencias.

En la siguiente sección primero se repasa la experimentación con las propiedades de algoritmos (corrección, eficiencia y optimalidad) y segundo se describe la experimentación y el método científico. Finalmente se exploran las dificultades encontradas por los alumnos a la hora de aprender los algoritmos.

2.2.1 Experimentación con las Propiedades de Algoritmos

En esta sección se describe la experimentación con las tres propiedades de algoritmos (corrección, eficiencia y optimalidad).

2.2.2.1 Experimentación con Corrección

Realizar varias pruebas puede revelar errores pero no asegura la corrección. Hay dos formas básicas para probar un algoritmo o programa: pruebas de caja negra y blanca.

Las pruebas de caja negra (Chin, et al., 2007) se centran en la funcionalidad del programa. Se trata de verificar que el programa hace lo que su especificación dice. Para una función, esto significa asegurar que la post-condición se satisface siempre que se cumpla su pre-condición. En este tipo de pruebas, el verificador no puede ver el interior del programa, solamente debe preparar conjuntos de datos de prueba suficientes para asegurar que las salidas del sistema son correctas para todas las entradas del sistema válidas. Además, el programa no debe bloquearse con entradas no válidas. En este tipo de pruebas se selecciona un subconjunto de casos de prueba que tiene la más alta probabilidad de detectar el mayor número de errores. Los siguientes son varios métodos utilizados para seleccionar un subconjunto efectivos de casos de prueba:

1. Partición de equivalencia. El dominio de entrada se divide en un número finito de clases de equivalencia tal que una prueba de un valor representativo de cada clase es equivalente a la de cualquier otro valor en su clase.
2. Análisis de valores límite (borde). Este método complementa la partición de equivalencia mediante la concentración de elementos en los bordes de cada clase.

3. Causa-Efecto gráfico. Los casos de prueba se convierten a una tabla de decisión. Esta tabla de decisión se genera a partir de un gráfico de causa-efecto que es una representación lógica de la funcionalidad que el programa está tratando de alcanzar.
4. Adivinación del error. Los casos de prueba se escriben sobre la base de una lista de condiciones propensas a errores. La generación de la lista se basa en la comprensión, la ejecución y lógica del programa, y por lo tanto depende en gran medida del conocimiento, la creatividad y la experiencia del desarrollador.

En las pruebas de caja blanca (Chin, et al., 2007), se analiza exhaustivamente el código del algoritmo. El verificador tiene pleno conocimiento del código para el programa y debe asegurarse de que todos y cada sección del código ha sido probado a fondo. El método de caja blanca se puede ver como una estrategia de diseño iterativo impulsado por análisis de cobertura de código. La prueba puede ser diseñada a lo largo de las siguientes etapas:

1. Elegir una meta de cobertura razonable.
2. Analizar la cobertura de casos de prueba actuales.
3. Si la cobertura total es inferior al objetivo predeterminado, estudiar los segmentos del código de baja cobertura.
4. Identificar e incluir los casos de prueba que pueden aumentar la cobertura.
5. Volver a ejecutar las pruebas y repetir los pasos 2-4 hasta que se alcance la meta de cobertura.

Existen diferentes medidas de cobertura, cada una de las cuales es cada vez más completa, pero más difícil de lograr:

- Cobertura de declaración. Asegurar que cada sentencia se ejecuta al menos una vez.
- Decisión de cobertura. Asegurar que todas las ramas de direcciones es atravesada y que cada punto de entrada se invoca al menos una vez.
- Cobertura de Condición. Asegurar que cada condición en una rama adquiere todos los resultados posibles al menos una vez y que cada punto de entrada se invoca al menos una vez.
- Cobertura de Decisión-Condición. Asegurarse de que tanto la cobertura de decisión y cobertura de condición son satisfechas.

- Cobertura de varias condiciones. Asegurar que todas las posibles combinaciones de resultados de condición en cada decisión se invocan al menos una vez.

Muchos errores y problemas (como el código innecesario o excesivamente complicado) pueden ser identificados con el método de caja blanca, incluso antes de que las pruebas se ejecuten realmente. Sin embargo, las pruebas de caja blanca tienen algunas limitaciones, como por ejemplo que no se puede demostrar que el programa cumple con sus especificaciones; tampoco puede identificar las rutas que faltan o datos dependientes del problema. Por lo tanto, hay que señalar que este método no produce por sí mismo una solución completa, pero se debe usar en combinación con las pruebas de caja negra.

2.2.2.2 Experimentación con Eficiencia

La experimentación con eficiencia es una forma de medir y comparar el rendimiento de un algoritmo. La comparación de tiempo de ejecución de algoritmos se puede utilizar para determinar el umbral para que un algoritmo tenga un mejor rendimiento que otro. Sin embargo, el rendimiento de algoritmos no tiene por qué limitarse al tiempo de ejecución, pero puede abarcar cualquier medida que se puede obtener en tiempo de ejecución. Velázquez y otros (2012) describen los requisitos principales de los sistemas para la experimentación con eficiencia:

1. Generación de datos de entrada de tamaño y características adecuadas.
2. Ejecución del algoritmo con una maquina abstracta (por ejemplo, un compilador o intérprete)
3. Instrumentación y contabilización de las operaciones relevantes.
4. Visualización de los resultados, lo cual es especialmente relevante cuando se comparan varios algoritmos o casos.

Existen varias publicaciones que describen experimentos con la eficiencia de un algoritmo, por ejemplo Laxer (2001) describió una experiencia donde se comparan algoritmos. Su objetivo era convencer a los alumnos que la eficiencia de un algoritmo $O(\log n)$ es mejor que uno de $O(n^2)$. La experiencia consiste en la comparación de los algoritmos de ordenación, la ordenación de burbuja, ordenación por selección y ordenación rápida. Los alumnos prepararon un informe de laboratorio escribiendo el análisis de sus resultados, discutiendo sobre cada algoritmo y analizando los

resultados. McCracken (1989) también diseño tres prácticas de laboratorio para el curso de algoritmos a base de experimentación con el objetivo de reforzar lo que se ha presentado en la clase teórica, poder plantear cuestiones para la siguiente clase y para que los alumnos aprendan mediante la comparación de ejecución de algoritmos.

2.2.2.3 Experimentación con Optimalidad

No se puede demostrar la optimalidad mediante la experimentación. Sin embargo, la experimentación es la base del descubrimiento científico dado que es una oportunidad a exponer a los alumnos el método científico. El aprendizaje por descubrimiento tiene muchas ventajas (Bruner, 1976) como son:

1. Mejorar la participación activa.
2. Promover la motivación.
3. Promover la autonomía, la responsabilidad y la independencia.
4. Desarrollar la creatividad y las habilidades para resolver problemas.

La experimentación puede ser utilizada para detectar la no-optimalidad o para reunir pruebas de la optimalidad de un algoritmo. Sólo sabemos de un método experimental y un sistema que apoye la experimentación de optimalidad de algoritmos voraces (Velázquez-Iturbide & Pérez-Carrasco, 2009a). En este método, se pide al alumno descubrir las funciones de selección que pueden caracterizar los algoritmos voraces óptimas para un problema de optimización dado. En general, el conjunto de funciones de selección posibles no es muy grande, ya que se limitan a valores de algunos parámetros o a valores derivados de ellos. Un ejemplo se encuentra en el apartado 3.1.1. Un experimento como el descrito es similar a los usados en ciencias, implicando los mismos conceptos:

- *Hipótesis*: Existe una o varias funciones de selección que permiten obtener siempre la solución óptima mediante un algoritmo voraz.
- *Refutación*: Si una función de selección no produce el mejor resultado con unos datos de entrada cualesquiera ya no es óptima. En consecuencia, queda refutada. Los datos de entrada sirven como *contraejemplo* de su condición de función de selección óptima.
- *Evidencia experimental*: Si una función de selección siempre produce el mejor resultado, no sabemos si es óptima, pero tenemos evidencia experimental de que es una candidata seria a

serlo. Cuantas más ejecuciones hayamos realizado con un resultado óptimo, mayor será la evidencia aportada.

- *Demostración formal*: Para garantizar que una función de selección es óptima, no basta con tener evidencia experimental, sino que tenemos que demostrarlo formalmente. El proceso también es parecido a la demostración de la corrección de un algoritmo. Podemos realizar *pruebas* que permitan encontrar errores, es decir, contraejemplos de su corrección. Cuantas más pruebas hagamos, mayor será la evidencia experimental de su corrección. Sin embargo, esta sólo será completa mediante una *verificación formal* del algoritmo.

2.2.2 Experimentación y el Método Científico

Según Zachary (1997), el proceso de resolución de un problema en las ciencias de computación implica 1) la identificación del problema; 2) planteamiento del problema en términos de un modelo matemático, 3) la identificación de un método de cálculo para resolver el modelo; 4) la aplicación del método de cálculo en un ordenador y 5) la evaluación de los resultados en el contexto de la aplicación, el método, el modelo y el problema. El método científico es un esquema lógico utilizado por los científicos para buscar respuestas a las preguntas planteadas en la ciencia. La experimentación es una parte integral del método científico. Podemos identificar cuatro etapas del método científico experimental (Denning, et al., 1989):

- Formular una hipótesis.
- Construir un modelo y hacer una predicción.
- Diseñar un experimento y recopilar datos.
- Analizar los resultados.

Se introdujo originalmente la ciencia experimental para el apoyo en el análisis de los algoritmos presentados en clases, dado que los alumnos encuentran difícil normalmente el análisis de los algoritmos porque implica más matemáticas de lo que están acostumbrados. Es más, los ejercicios de laboratorio que implican experimentos científicos tienen una serie de beneficios (Baldwin, 1992):

- Conservar la virtud tradicional de mostrar a los alumnos realizaciones concretas de los conceptos de clase abstracta.

- Permitir a los alumnos explorar por su cuenta. Los alumnos están aprendiendo a aprender por su cuenta.
- Ayudar a desarrollar habilidades de comunicación mediante el desarrollo los informes de laboratorio que deben explicar claramente los procedimientos experimentales y conclusiones.

Conscientemente formar una hipótesis y predecir un resultado antes de hacer cualquier otra cosa es un hábito valioso que obliga a pensar en lo que debería suceder y por qué. Incluso una predicción incorrecta es valiosa, ya que obliga a uno a preguntarse por qué el resultado inesperado ocurrió y así corregir malentendidos o métodos erróneos. Con la utilización del método científico en las ciencias de la computación, se espera alcanzar tres objetivos importantes (Moore, et al., 1993):

- 1) Perfeccionar la habilidad de “pensar críticamente“
- 2) Atraer los alumnos a la informática.
- 3) Establecer una base para clases posteriores.

Fenwick (2002) describió una experiencia elaborada a base la realización de tres practicas con ejercicios de programación con el objetivo de introducir la experimentación científica a las ciencias de computación. Esta experiencia reveló una mejora de las habilidades de pensamiento empírico y crítico de los alumnos. En las dos primeras prácticas, se daba a los alumnos el algoritmo y la implementación y se debían enlazarles (emparejar cada implementación con su algoritmo) y en la tercera práctica tenían que implementar un algoritmo y realizar un estudio y análisis experimental.

Otra experiencia (Matocha, 2002) describió un conjunto de experimentos de laboratorio para que los alumnos practicasen las habilidades de escritura técnica y el uso del método científico dado su importancia para los alumnos. En estas prácticas, los alumnos debían formular la hipótesis de diseño de experimentos, dar pruebas de hipótesis y analizar los datos. Además siguieron un conjunto de directrices para el desarrollo de sus informes de laboratorio: 1) Describe lo que observas, 2) Discuta las ventajas de un enfoque sobre otro, 3) Justifica tu respuesta y 4) Justificar la selección de las estructuras de datos o algoritmo.

2.2.3 Dificultades de los Alumnos

Según Winslow (1996), la resolución de problemas se puede dividir en cuatro pasos:

1. Entender el problema.
2. Determinar la forma de resolver el problema:
 - A. en alguna forma y
 - B. en forma compatible con la computadora.
3. Traducir la solución al lenguaje de programación.
4. Probar y depurar el programa.

Sin embargo, según Haberman y sus colegas (2005), los alumnos casi no distinguen diferentes elementos del proceso de desarrollo de un algoritmo, como el algoritmo mismo, sus decisiones de diseño subyacentes y el análisis de resultados. Además subrayan el papel fundamental del profesor en el establecimiento de normas y conceptos de programación, por ejemplo enseñar a los alumnos los principios y las estrategias que se deben aplicar en el diseño de un programa y ejemplos de los procesos de diseño completos en lugar de mostrarles solamente productos finales. También debe indicarse (a los alumnos) que las explicaciones y justificaciones son elementos inherentes en la solución de un problema algorítmico.

El objetivo es que los alumnos puedan desarrollar algoritmos o programas informáticos que cuenten con las características fundamentales de exactitud, sencillez, claridad y eficiencia. Por otro lado, no se debe omitir ningún componente de una solución algorítmica (algoritmo, prueba de corrección y análisis de complejidad). Por eso, se debe exigir a los alumnos producir una solución algorítmica a un problema dado con su prueba de corrección y análisis de complejidad (Armoni, 2006). Es más, se les debe exigir la demostración de la exactitud de su programa y dar una explicación (a los demás) por qué el programa es correcto (Kolikant & Pollack, 2010). Este razonamiento y verificación de la corrección puede implicar en particular el fomento de la exploración de las pruebas en contra de la hipótesis intuitiva del alumno (Ginat, 2001).

Kolikant (2005) señaló que los alumnos en general entienden corrección como una propiedad relativa del programa y toleran errores. Ginat (2003) presentó un estudio de un fenómeno de error común recurrente que se produce en una variedad de ocasiones para resolver problemas algorítmicos de optimización y un enfoque de aprender de estos errores. Los alumnos se dieron cuenta de la "fragilidad" de la intuición y el papel esencial de rigor mediante la búsqueda de pruebas en contra de sus soluciones a través de modificaciones radicales de sus soluciones, seguido por la argumentación

rigurosa y la reflexión sobre todo el proceso. Ginat (2007), años más tarde, alegó que los errores de los aprendices se deben a diseños precipitados que no se basan en observaciones rigurosas y consideraciones. Los aprendices a menudo tienden a examinar muy pocos casos y se vuelven demasiado rápido para el diseño de programas basado en estos casos, además tienen la tendencia de seguir sus pensamientos iniciales, evitando el desvío a direcciones alternativas. Con este análisis, se utilizaron los diseños apresurados y parches inútiles de los alumnos para la elaboración de rigor y la enseñanza del proceso de diseño apropiado como forma de aprender de los errores. Ginat (2008), en otro trabajo, propuso un esquema de “aprender por errores” que encapsula dos temas principales: aprendizaje de las tendencias erróneas comunes en el diseño de algoritmos y el uso de soluciones algorítmicas erróneas para el razonamiento creativo. Este enfoque incluyó dos componentes 1) una colección de tareas algorítmicas (que no son ejercicios comunes) y 2) las actividades de clase. Cada actividad termina con una discusión en clase que incluye la reflexión sobre: direcciones de soluciones adecuadas y no adecuadas y el pensamiento divergente que produce los resultados deseados. Este enfoque resultó eficaz y les gustó a los alumnos.

Con respecto a los problemas de optimización, Velázquez (Aceptado) identificó varios malentendidos sobre los conceptos básicos de optimización que subyacen en los algoritmos voraces. Por ejemplo, que los alumnos no comprueban la optimalidad de un algoritmo para todas las entradas válidas, sino que aceptan tener diferentes algoritmos óptimos para diferentes datos de entrada. Esta tolerancia es consistente con otros hallazgos (Kolikant, 2005) sobre la comprensión superficial de otras propiedades de los algoritmos: la corrección y la eficiencia. Por lo tanto, los alumnos no se exigen que un algoritmo sea válido para todos los datos de entrada. Es más, muchos alumnos no podían aceptar que varias funciones de selección podrían ser óptimas y esto implica una concepción errónea de optimalidad.

2.3 Sistemas Interactivos para el Aprendizaje de Algoritmos

En esta sección se revisan los sistemas interactivos para el aprendizaje de los algoritmos. En primer lugar, se recogen los sistemas correctores que son herramientas de calificación automática. En segundo lugar, se repasan los sistemas de visualización de algoritmos y por último se enumeran algunos sistemas de visualización de programas.

2.3.1 Sistemas Correctores

En la siguiente sección, en primer lugar, se definen los sistemas de calificación automática y se enumeran sus ventajas. En segundo lugar, se recogen los sistemas correctores.

2.3.1.1 Sistemas de Calificación Automática

Los sistemas de calificación automática (*automatic grading system*) son herramientas que hacen uso de los avances en la tecnología Web y adoptan enfoques de prueba cada vez más sofisticados. La evaluación automatizada puede producir beneficios para los profesores y alumnos. Douce y sus colegas (2005) describen una serie de sistemas de evaluación automática influyentes. En su artículo argumentan una serie de ventajas e inconvenientes del uso de estos sistemas. Por ejemplo, que ahorran mucho tiempo para el profesor, dado que la evaluación de las tareas de programación es una tarea difícil y consume mucho tiempo, entonces el profesor puede dedicar más tiempo dando orientación a los alumnos y explicando los conceptos que encuentran difícil de entender. Otra ventaja es que los evaluadores humanos son falibles, mientras que los evaluadores automáticos pueden dar respuestas totalmente objetivas. De las inconvenientes se encuentra las restricciones que se aplican cuando se evalúa de forma automática. Un sistema de evaluación no puede identificar una solución innovadora o creativa del alumno y por lo tanto no la puede premiar (subir nota). Siempre habrá una necesidad de los evaluadores humanos. Estos sistemas pueden proporcionar apoyo para el profesor permitiendo identificar las ideas erróneas potenciales o los problemas de los alumnos más fácilmente. En resumen, se debe combinar estos sistemas con la evaluación humana.

Por otro lado, Cheang y sus colegas (2003) identifican tres componentes principales que intervienen en la calificación de las tareas de programación:

- 1) Exactitud. El componente más importante al calificar una tarea de programación debe ser su corrección. Un programa es correcto si muestra el comportamiento correcto de Entrada/Salida para todas las entradas en el dominio del espacio del problema.
- 2) Eficiencia. Un programa es eficiente si desempeña sus actividades sin consumir demasiado tiempo de procesamiento y espacio de memoria.

- 3) **Mantenibilidad.** Un programa es fácil de mantener si el código es fácil de entender empleando nombres descriptivos de variables, comentarios, sangría y programación modular.

2.3.1.2 Revisión de los Sistemas Correctores

Podemos distinguir dos tipos de sistemas, sistemas que dan apoyo a la gestión de los concursos de programación y que están diseñados para evaluar a los concursantes de las competiciones de programación y sistemas para el aprendizaje de programación que están diseñados para los cursos de programación y que permiten a sus usuarios practicar y evaluar sus programas.

En primer lugar, de los sistemas que apoyan la gestión de concursos de programación, encontramos el entorno **PC2** (Anon., 2013) que está diseñado para apoyar las competiciones de programación y tiene capacidades para la gestión individual y de concursos. El sistema también soporta el modo “jueces automáticos”, donde se juzga mediante la herramienta sin la necesidad de jueces humanos. PC2 proporciona un mecanismo para que los concursantes presenten solicitudes de aclaración y consultas a los jueces. Los concursantes presentan sus programas. Los jueces pueden volver a compilar el programa presentado, ejecutarlo, ver el código fuente y los resultados de la ejecución y enviar una respuesta a los concursantes.

Del mismo modo, el sistema **Mooshak** (Leal & Silva, 2003) actúa como un administrador de concurso completo y como un juez automático para concursos de programación. Mooshak se aplica además para el aprendizaje de programación. Tiene capacidades de evaluación automática para ayudar a los jueces humanos en la evaluación de programas. Los concursantes presentan programas de código, hacen preguntas a los jueces y acceden a todas las preguntas formuladas por los participantes y sus correspondientes respuestas dadas por los jueces. Además permite el acceso a la lista de todas las presentaciones y las notas correspondientes, imprimir los programas de código de desarrollo y visualizar en el navegador descripciones del problema. Este sistema es una herramienta automática que es capaz de evaluar la corrección de los programas a base de un conjunto predeterminado de pares de Entrada/Salida. Cuenta con una interfaz basada en la web para los profesores, alumnos, usuarios invitados y el administrador del sistema. García y Alemán (2009) detallaron una experiencia innovadora en el curso de Algoritmos y Estructuras de Datos usando Mooshak. En esta experiencia ha sido remplazada por completo la evaluación tradicional del examen

final con una serie de actividades, muchas de ellas con Mooshak, obteniendo dos beneficios: la motivación de los alumnos para participar en el actividades propuestas para evitar el examen final y la evaluación del trabajo del alumno durante el curso, en lugar de en un único examen, consiguiendo la objetividad de la evaluación. Con Mooshak, los alumnos tienen que ser mucho más riguroso en el desarrollo de sus programas obteniendo retroalimentación inmediata del sistema. Como ventaja está el ahorro de tiempo y trabajo por parte del profesor con el uso de un sistema de evaluación automática aunque los conceptos importantes como la robustez y la legibilidad deben ser puntuadas manualmente por los profesores. Además, la usabilidad de Mooshak ha sido evaluada positivamente.

Online Judge (Cheang, et al., 2003) es otro sistema en línea para concursos de programación que comprueba la corrección y eficiencia de un programa ejecutando los programas presentados. La corrección depende de si la salida producida coincide con las respuestas pre-especificados. La eficiencia de una presentación depende de si el programa es capaz de producir su salida dentro del límite de tiempo y recursos de memoria. Los alumnos deben presentar sus soluciones como programas que se pueden ejecutar. Los programas se compilan y ejecutan en el servidor, la salida del programa presentado se compara con la salida real. El sistema permite detectar casos de plagio. Sin embargo, los comentarios del sistema a los alumnos no son concisos. Además Online Judge no es capaz de manejar varios cursos a la vez y el sistema no puede determinar la mantenibilidad de un programa presentado.

En segundo lugar, existen varios sistemas que se utilizan para los cursos de aprendizaje de programación o para mejorar las habilidades de programación de los alumnos o entrenar a los concursantes dado que practicar es uno de los pasos más importantes en el aprendizaje de la programación. Por ejemplo **UVa Onlinejudge** (Revilla, et al., 2008) es un repositorio de problemas de programación que tiene una colección grande y creciente de problemas de programación (más de 4.300 problemas). Proporciona un juez automatizado en línea 24 horas que interactúa con los usuarios a través de los e-mails. Es un servidor que contiene la descripción de los problemas de diferentes concursos. Un usuario, desde cualquier parte del mundo, puede registrar en el sistema de forma gratuita y resolver tantos problemas que quiera. Incluso puede enviar tantas soluciones como él quiera, hasta que reciba información satisfactoria, no sólo acerca de la corrección de su programa, sino también acerca del tiempo que el código tardó en ejecutarse. El usuario recibe una evaluación más completa del sistema, no sólo indicando que el problema es correcto (o no) pero además

incluyendo la calificación de la calidad de las soluciones. Esto puede variar desde una solución no significativa (menos de 50% de la corrección) a una solución completamente correcta (100%). También permite analizar la eficiencia de un programa.

CourseMarker (Higgins, et al., 2003)(Higgins, et al., 2002) es otra herramienta para cursos de programación. El alumno puede seleccionar el curso, el tema y el ejercicio que quiere completar. La herramienta permite comprobar el diseño del programa presentado como la sangría, elección y duración de identificadores y el uso de comentarios. Todos los parámetros de la tipografía se pueden personalizar en base del ejercicio. CourseMarker permite a los alumnos volver a presentar su solución tantas veces como especifica el desarrollador del ejercicio. La evaluación se realiza mediante un mecanismo automatizado que analiza el programa a través de una serie de criterios con el énfasis en la exploración de diseño del programa. El sistema puede detectar casos de plagio. Una de sus novedades más interesantes es su apoyo a una impresionante variedad de lenguajes, incluyendo Prolog, SQL y FORTRAN. Incluso se puede ver la información acerca del progreso de los alumnos. Además se ha comprobado su usabilidad positivamente.

El sistema **ASAP** (Douce, et al., 2005) de la Universidad de Kingston es una herramienta para el aprendizaje de lenguaje de programación Java. El sistema permite a los profesores ver la solución de los alumnos a través de un servicio Web y evaluar automáticamente el código del alumno con comentarios apropiados. Cuando un alumno envía una tarea, su programa se califica y se evalúa. Un documento XML, que contiene los comentarios acerca del programa, una descripción de las pruebas y una calificación final, se genera automáticamente. Los comentarios proporcionados por el sistema pueden ser muy simples o más detallados (en el caso de un resultado incorrecto, se proporcionan sugerencias a los alumnos para la respuesta correcta). Incluso permite la generación automatizada de preguntas objetivas para poner a prueba los resultados de aprendizaje y la detección de plagio.

Jutge.org (Giménez, et al., 2012) es un juez de programación educativa en línea que está diseñado tanto para los profesores como los alumnos. Los alumnos pueden inscribirse en los cursos del profesor y en los cursos públicos para practicar. El repositorio de problemas en Jutge.org está organizado por temas y clasificado por dificultad. Los profesores pueden adaptar sus cursos de programación utilizando una gran base de 800 problemas y 22 lenguajes de programación de

acuerdo a sus necesidades y preferencias. En Judge.org, una solución del alumno no debe coincidir exactamente con los resultados producidos por la solución de Judge.org. En el proceso de validación de una solución del alumno no se da por incorrecta si no tiene un formato estricto de la salida o si las palabras están separadas por un espacio o una línea en blanco. Además el orden de la salida también es irrelevante, siempre que contenga todo lo que debe contener. Es más, cuando la salida de un problema es un número flotante (*float*), una aproximación puede ser tolerada. Finalmente, en los problemas de optimización, las soluciones no son únicas. Judge.org comprueba la salida para este tipo de problemas utilizando correctores externos. Primero se calcula la solución del problema utilizando su propia solución y luego se comprueba si es el mismo resultado introducido por el alumno. Las principales características de Judge.org para los profesores son la automatización del proceso de corrección, la creación y el mantenimiento de sus propios cursos, la gestión de sus listas de tareas y el acceso a la información acerca del progreso de los alumnos.

AlgoVista (Collberg, et al., 2004)(Collberg, et al., 2002) también puede ser utilizado como una herramienta para la verificación de programas. Cada algoritmo tiene vinculado un corrector que es el encargado de verificar la exactitud de los resultados que calcula. La base de datos almacena tanto las descripciones de problema como su verificación o “*checklets*”. Además, es un motor de búsqueda basado en la Web que ayuda a los alumnos a encontrar algoritmos e implementaciones que resuelven problemas específicos. Para buscar un algoritmo determinado, un usuario simplemente escribe la consulta mediante un editor sobre la base de un navegador Web. El resultado de la búsqueda es una lista de enlaces a recursos Web que describen o proporcionan implementaciones del algoritmo. El sistema se puede utilizar para crear tareas de programación interesantes en clases de algoritmos.

2.3.2 Sistemas de Visualización de Algoritmos

En esta sección se define la visualización y la animación de algoritmos y se enumeran los sistemas de visualización de algoritmos.

2.3.2.1 Visualización y Animación de Algoritmos

El concepto de “algoritmo” es uno de los temas básicos y más importantes en la enseñanza de computación, sin embargo los alumnos les resulta difícil de entender por ser un concepto muy

abstracto. Por eso, las técnicas de visualización y animación pueden ser útiles como ayuda visual para entender la eficiencia de los algoritmos.

Según Stephan Diehl (2007) la visualización del software es “la visualización de artefactos relacionados con el software” y el proceso de desarrollo es “la visualización de la estructura, el comportamiento y la evolución del software”.

1. La estructura: se refiere a las partes y las relaciones estáticas del sistema, es decir, aquellos que pueden ser calculadas o inferirse sin ejecutar el programa.
2. Comportamiento: se refiere a la ejecución del programa con los datos reales y abstractos.
3. La ejecución: puede ser visto como una secuencia de estados del programa, donde un estado del programa contiene tanto el código actual y los datos del programa.
4. La evolución: se refiere al proceso de desarrollo del software.

Asimismo, la animación de algoritmos es la visualización del comportamiento de un algoritmo (Diehl, 2007). La ejecución de un algoritmo conduce a una secuencia de estados. Cada paso de los resultados del algoritmo en una transición de un estado a otro. La animación de algoritmos transforma cada estado en una representación visual (imagen) y muestra las transiciones como animaciones entre estas imágenes. La animación de algoritmos tiene varias ventajas como:

1. El aprendizaje y la enseñanza: los profesores visualizan algoritmos para explicar a sus alumnos.
2. Diseño: los desarrolladores visualizan algoritmos para mejorar la comunicación de las ideas a otros expertos.
3. Optimización: los desarrolladores visualizan algoritmos para entender mejor cómo funcionan y encontrar posibilidades de reforzarlos.
4. Depuración: los programadores usan visualizaciones para encontrar errores en sus programas.

Shaffer y sus colegas (2010) presentan el estado de la cuestión de los sistemas de visualización de algoritmos. En su trabajo indican la dificultad de crear y desplegar sistemas de visualización eficaces. Además, los de alta calidad se pueden perder entre los muchos de menor

calidad. Urquiza y Velázquez (2009) también describen las experiencias educativas del uso de los sistemas de visualización de programas/algoritmos. En su trabajo incluyen las siguientes recomendaciones para el diseño de experiencias de aprendizaje de visualización:

- Las animaciones deben tener texto adicional o contenidos narrativos.
- Permitir a los alumnos cambiar los datos de entrada es una tarea más activa.
- Cuando los alumnos construyen sus propias animaciones, la interfaz de la construcción es muy importante.
- Cuando los alumnos responden a preguntas durante una animación, también deben estar provistos de contenido textual adicional. Retroalimentación explícita también es importante.
- Animar los alumnos a producir sus propios textos o contenidos narrativos también es positivo.

2.3.2.2 Clasificación de los Sistemas de Visualización

Hemos clasificado los sistemas de visualización para el aprendizaje de algoritmos de optimización en tres tipos: sistemas de visualización y animación de algoritmos, sistemas de visualización y verificación (de la comprensión del algoritmo) y sistemas de visualización y comparación de algoritmos.

En primer lugar, los sistemas de visualización y animación permiten crear las visualizaciones y ver las animaciones para el aprendizaje de los algoritmos de optimización. En esta categoría, podemos citar el sistema **GraphsJ** (Costa, et al., 2010) que fue desarrollado para ayudar a los alumnos a aprender los algoritmos de grafos. GraphsJ proporciona una interfaz independiente y una biblioteca que cualquiera puede utilizar para el desarrollo personalizado de algoritmos. La interfaz de usuario permite realizar la ejecución "paso a paso" y suministra información detallada que ayuda a los alumnos en su aprendizaje básico de algoritmos de grafos. Los siguientes algoritmos fueron implementados en GraphsJ para la solución de cuatro básicos problemas de la teoría de grafos: el árbol del recubrimiento del coste mínimo, los caminos más cortos, el flujo máximo y el camino crítico.

VisualGraph (Lucas, et al., 2003) proporciona una biblioteca para la animación de algoritmos de grafos. No cuenta con una interfaz de usuario propia. Para ver la animación, se requiere el uso de

la herramienta Animal para la visualización del grafo. AnimalScript es el lenguaje para la implementación de los gráficos.

El sistema **INTEGER** (Ramani & Rama Rao, 1994) es una animación gráfica interactiva para resolver problemas de programación entera y encontrar el árbol de solución utilizando la técnica de ramificación y poda (*Branch and Bound*). El sistema se sometió a una evaluación de eficacia educativa que mostró que la herramienta tiene una influencia positiva en las calificaciones de los alumnos que utilizaron el sistema.

El sistema **EDAapplet** (Almeida, et al., 2003) es una aplicación Web orientada a la enseñanza/aprendizaje de la algorítmica. EDApplets ofrece visualización del comportamiento del algoritmo mediante animaciones gráficas, destacando la línea del pseudocódigo, la visualización contempla, la ejecución paso a paso de los algoritmos para entradas proporcionadas por los alumnos y la presentación de trazas del código. Los alumnos pueden contestar a preguntas teóricas tras terminar con la herramienta. Además, permite ilustrar estructuras de datos, el algoritmo de vuelta atrás, divide y vencerás, programación dinámica, recursividad, ramificación y poda y también las técnicas de ordenación utilizando algoritmos conocidos como *QuickSort*, *MergeSort* o *SelectionSort*.

AlgoViz (Anon., 2009), como hemos mencionado anteriormente, es sin duda uno de los portales más famosos de las animaciones de algoritmos que incluye una serie de animaciones para el algoritmo de Dijkstra, Kruskal y de Prim, códigos de Huffman y otros algoritmos.

En segundo lugar, existen otros sistemas de visualización que verifican la comprensión de los alumnos mediante la comprobación de sus respuestas o soluciones (pasos realizados) y que incluso proporcionan mensajes de aclaración a los alumnos. Dentro de esta categoría están los sistemas que están diseñados para algoritmos de grafos como **PathFinder** (Sánchez-Torrubia, et al., 2009) que es un sistema que apoya el aprendizaje de algoritmos voraces, en particular, el algoritmo de Dijkstra. Este sistema permite la ejecución del algoritmo paso a paso suministrando mensajes de explicación en cada paso, además permite verificar la solución del alumno. El sistema también devuelve mensajes de error personalizadas. La ejecución tiene tres modos de ejecución: dos modos de funcionamiento interactivo (verificación paso a paso y verificación Iterativa) y una opción de ejecución directa capaz de proporcionar directamente el resultado final. En cada paso realizado por el alumno (en el modo de verificación paso a paso), para cada iteración del algoritmo el alumno debe

actualizar primero los nodos fijos y no fijos, comprobar si es correcto o no y finalmente actualizar distancias y comprobar de nuevo la corrección de entrada. El modo de verificación Iterativa está dirigido a los alumnos más avanzados donde el alumno debe realizar toda una iteración antes de verificar. Por cada verificación (en ambos modos de funcionamiento interactivos), cuando alguna de las actualizaciones no es correcto, el panel de mensajes muestra un mensaje de error señalando el problema y proporcionando sugerencias para la corrección. Además proporciona la opción de crear, editar, guardar y recuperar un gráfico. El sistema **GRAPHS** (Sánchez-Torrubia, et al., 2010) es un sistema parecido al PathFinder. Es un entorno concebido para mejorar el aprendizaje activo y autónomo de varios algoritmos de grafos mediante la simulación visual. Cuenta con las mismas funcionalidades de PathFinder, pero además los alumnos se exponen a preguntas adaptadas al nivel de dificultad del algoritmo y diseñados para que piensen acerca de otros aspectos del funcionamiento del algoritmo.

DIDAGRAPH (Dagdilelis & Satratzemi, 1998) es un entorno para la visualización y exploración de algoritmos de grafos. Esta herramienta cuenta con pocos algoritmos (Dijkstra, Kruskal y otro más). DIDAGRAPH permite acompañar las animaciones de ejecución de algoritmos a través de la visualización de grafos con una descripción de lenguaje de alto nivel del algoritmo. También suministra mensajes de explicación a los alumnos.

IAPPGA (Wu, 2005) proporciona a los alumnos un ambiente para estudiar algoritmos de grafos. El alumno seleccione el conjunto de datos de entrada y automáticamente se visualiza la solución paso a paso utilizando el algoritmo de Edmonds-Karp acompañado con mensajes de explicación de la ejecución del algoritmo. Si un usuario proporciona los datos de entrada incorrectos, el programa proporciona un mensaje de error y le pedirá al usuario que los cambie.

PILOT (Bridgeman, et al., 2000) es un sistema de visualización de algoritmos. El alumno puede usar el sistema en tres diferentes modos de interacción. En el modo "Aprendizaje", el alumno ejecuta el algoritmo, en cada paso el sistema comprueba si ese paso es correcto. En caso contrario, se genera automáticamente un mensaje de explicación y en algunos casos una animación en pantalla. En el modo de "Práctica con exámenes", el alumno crea su solución sin retroalimentación y luego se corrige la solución y se le asigna una nota. El modo de "examen" está diseñado para que los alumnos lo utilicen para presentar una solución para la nota. La solución y la nota obtenida se registran

automáticamente. El sistema incluye el problema del árbol de recubrimiento de coste mínima (MST) usando los algoritmos Kruskal y Prim.

En la misma categoría, hemos recogido sistemas de visualización de algoritmos generales que incluyen algoritmos de optimización (p.ej. el problema de camino más corto), por ejemplo está el sistema hipertexto de visualización denominado **HalVis** (Hansen, et al., 2002) que proporciona un entorno interactivo para el aprendizaje de algoritmos que atrae la atención visual y cognitiva del alumno a través de animaciones atractivas acompañados con explicaciones, diagramas estáticos, ejemplos interactivos para proporcionar información contextual y preguntas que estimulan el pensamiento y fomentan la auto-explicación. HalVis plantea periódicamente preguntas al alumno. Hay dos tipos de preguntas: "Tickler" son preguntas abiertas que aparecen en un orden aleatorio, pero siempre en un contexto apropiado. Su objetivo es centrar la atención de los alumnos en cuestiones específicas, desafiar su entendimiento y promover el auto-explicación y preguntas de tipo "opción múltiple", en este caso, el sistema proporciona retroalimentación inmediata. Este sistema incluye varias técnicas de ordenación como *QuickSort*, *MergeSort* o *SelectionSort* y el problema de camino más corto (*Shortest path*). La línea del pseudocódigo que está siendo ejecutado en la animación se destaca, incluso suministra breves mensajes explicativos acerca de las operaciones que se están ejecutando y una vista de los valores de las variables importantes utilizados por el algoritmo. HalVis también demostró ser muy eficaz en ayudar a los alumnos a aprender algoritmos.

Otros sistemas, en esta misma categoría, incluso permiten generar animaciones, como es el caso de **TRAKLA2** (Malmi, et al., 2004) y **JHAVE** (Naps, 2005). El sistema TRAKLA2 es una interfaz diseñada para visualizar el comportamiento de los algoritmos con mensajes de explicación y animación (es un sistema general de algoritmos que incluye el algoritmo Dijkstra y otros más). TRAKLA2 es un entorno visual que permite la evaluación de la simulación de algoritmos y estructuras de datos. Este sistema distribuye ejercicios adaptados individualmente a los alumnos y los evalúa automáticamente. Un alumno puede manipular directamente la representación visual de las estructuras de datos subyacentes a los que se aplica el algoritmo, luego puede pedir la verificación de su solución, entonces el sistema le entrega un mensaje avisando de los pasos correctos de la simulación de su solución. Por ejemplo, Trakla2 pide al alumno agregar aristas hasta construir un árbol de caminos más cortos en el orden en que el algoritmo de Dijkstra lo haría. JHAVE es una herramienta que presenta un innovador sistema de visualización de algoritmos que

proporciona un motor que incluye enlaces a documentación externa y preguntas interactivas a los usuarios para responder mientras ven la animación. El usuario una vez conectado selecciona un algoritmo, el conjunto de datos de entrada y automáticamente se visualiza el algoritmo. Además proporciona preguntas “*stop-and-think questions*” sobre el algoritmo que aparecen durante el curso de la visualización (acerca de la visualización que el sistema presenta). En las pregunta se pide típicamente predecir lo que van a ver en la animación del algoritmo. Entonces, el sistema verifica la respuesta del alumno. También incluye algoritmos de optimización como es el caso de Dijkstra.

En tercer lugar, están los sistemas de visualización que permiten la comparación de algoritmos vía la visualización. Podemos citar el sistema **AI-Search** (Ciesielski & McDonald, 2001) que es una herramienta de animación basada en los problemas de 8-PUZZLE que se utilizan en casi todos los libros de texto para ilustrar los conceptos de algoritmos de búsqueda. El sistema contiene todos los algoritmos de búsqueda generalmente descritos en libros de texto de introducción a la inteligencia artificial incluida la búsqueda en profundidad, la búsqueda en anchura, profundización iterativa, algoritmo de búsqueda A *, búsqueda voraz y ramificación y poda (*branch and bound*). El sistema cuenta con dos modos: Modo “*single step mode*” que facilita el aprendizaje de los pasos individuales de los distintos algoritmos de búsqueda mientras que el modo “*burst mode*” facilita el aprendizaje del comportamiento cualitativo y comparación de algoritmos. Los resultados de la evaluación de usabilidad indicaron que los alumnos creen que el uso del sistema aumenta su comprensión de los algoritmos de búsqueda. Además, en la evaluación empírica ambos grupos (tanto los alumnos que usaron el sistema como los que no) obtuvieron resultados similares. Sin embargo, los resultados han sido mejor en el grupo que usó el sistema en las cuestiones relativas a rendimiento cualitativo de los algoritmos de búsqueda.

EVEGA (Khuri & Holzapfel, 2001) es una herramienta potente para la creación y edición interactiva de grafos y algoritmos de animación. Incluye la animación del algoritmo para el problema del flujo máximo. Los alumnos pueden escribir sus propias implementaciones de algoritmos en Java permitiendo ver los resultados de sus programas en formato gráfico. Además las ejecuciones de varios algoritmos se pueden comparar utilizando gráficos de rendimiento generados por la herramienta. Los profesores también pueden ver los errores en la implementación del algoritmo más fácilmente. Por otra parte, el sistema desarrollado por (Chen, et al., 2006) describe un algoritmo Benchmark que experimenta con la eficiencia. Consiste en comparar el espacio de

memoria, el tiempo de ejecución y la salida de ciertas funciones de evaluación entre varios algoritmos. Esta comparación se basa típicamente en el rendimiento de datos brutos.

2.3.3 Sistemas de Visualización de Programas

Existen sistemas de visualización de programas para el aprendizaje de la programación. Podemos citar, la herramienta **Ville** (Kaila, et al., 2009) que es un sistema de visualización de programas para la enseñanza de conceptos básicos de programación que soporta múltiples lenguajes de programación. Su principal objetivo es ilustrar los cambios durante la ejecución con diversos medios gráficos y de texto. Los ejercicios evaluados automáticamente de Ville se pueden integrar fácilmente como parte de un curso de programación. Del mismo modo, el sistema **Jeliot3** (Moreno, et al., 2004) es un sistema de visualización de programas diseñado para visualizar las características básicas del lenguaje Java en los cursos introductorias de programación. Está enfocado al paradigma de la programación orientada a objetos y permite a los alumnos visualizar inmediatamente sus propios programas. Jeliot3 es una extensión que permite interactuar con **BlueJ**. Además permite el uso de la herramienta directamente desde una página Web. **jGRASP** (Cross & Hendrix, 2007) es una herramienta que proporciona capacidades de visualización tanto estáticas como dinámicas. Sus visualizaciones son poderosas porque proporciona múltiples visualizaciones del código fuente: los objetos de bajo nivel y las de alto nivel. Es un entorno de programación que soporta compilación, depuración y ejecución con capacidad de visualización. **VIP** (Virtanen, et al., 2005) es otro sistema de visualización de programas que permite asociar explicaciones textuales a las localizaciones del código fuente. El usuario puede escribir pseudocódigo y observar su simulación. Con este entorno es posible ejecutar cada línea del pseudocódigo paso a paso y observar la evolución de los valores de variables y también la salida del programa. **Verificator** (Radošević, et al., 2009) es una herramienta de software destinada a ayudar los alumnos a adquirir fácilmente más conocimientos de programación y mejorar el proceso de enseñanza de programación. El objetivo principal del verificador es empujar a los estudiantes a tomar los puntos de control durante el desarrollo del programa, por lo que no pueden seguir escribiendo su programa hasta que su sintaxis es correcta. Verificator es una interfaz de aprendizaje de programación que tiene herramientas para el análisis de la sintaxis y la depuración, incluyendo la búsqueda de símbolos de cierre de las estructuras de un programa. El sistema **WinHIPE20** (Pareja-Flores, et al., 2007) es un entorno funcional de programación con capacidades de visualización, donde se visualizan expresiones funcionales

incluyendo explicaciones textuales que contienen ambas estructuras de datos (árboles o listas) y construcciones de lenguaje.

2.4 Sistemas CSCL

En esta sección se describen las teorías educativas, el paradigma de Aprendizaje Colaborativo Soportado por Computador y los sistemas CSCL para la enseñanza de algoritmos y programación.

2.4.1 Teorías Educativas

La distinción básica entre las tres grandes corrientes (conductismo, cognitivismo y constructivismo) radica en la forma en que se concibe el conocimiento. Para el conductismo, el conocimiento consiste fundamentalmente en una respuesta pasiva y automática a factores o estímulos externos que se encuentran en el ambiente. El cognitivismo considera el conocimiento básicamente como representaciones simbólicas en la mente de los individuos. Por último, el constructivismo, concibe al conocimiento como algo que se construye, algo que cada individuo elabora a través de un proceso de aprendizaje (Ertmer & Newby, 1993).

En primer lugar, el conductismo define el aprendizaje como la adquisición de nuevos comportamientos. El conductismo opera sobre un principio de "estímulo-respuesta". El aprendizaje se logra cuando se demuestra o se exhibe una respuesta apropiada a continuación de la presentación de un estímulo ambiental específico (Ertmer & Newby, 1993). Sus principales procesos son conocidos como condicionamiento clásico y condicionamiento operante:

1. Condicionamiento clásico por Pavlov: se produce cuando un cierto estímulo producirá una respuesta específica.
2. Condicionamiento operante por Skinner: es el proceso de aprendizaje en el que una conducta o comportamiento deseado es fortalecido o debilitado por una recompensa o refuerzo.

En segundo lugar, según Piaget, el desarrollo cognitivo es una reorganización progresiva de los procesos mentales como resultado de la maduración biológica y la experiencia del medio ambiente. La adquisición del conocimiento se describe como una actividad mental que implica una codificación interna y una estructuración por parte del estudiante. Se observan cambios en el

comportamiento y se utilizan como indicadores de lo que está sucediendo dentro de la mente del alumno (Ertmer & Newby, 1993).

Finalmente, el aprendizaje constructivo se forma construyendo nuestros propios conocimientos desde nuestras propias experiencias a través de la interacción con el ambiente. Los individuos construyen nuevos conocimientos a partir de experiencias y a través de procesos de acomodación y asimilación. En este caso la asimilación es el proceso por el que el sujeto interpreta la información que proviene del medio en función de sus esquemas o estructuras mentales ya existentes, sin embargo, la acomodación supone no sólo una modificación de los esquemas previos, sino también una nueva asimilación o reinterpretación de los conocimientos anteriores (Ertmer & Newby, 1993).

Las técnicas del aprendizaje constructivista o los modelos instruccionales que potencian el aprendizaje centrado en el usuario son propuestas más abiertas: en primer lugar, PBL (*Problem Based Learning*) consiste en desafiar la capacidad de los alumnos para utilizar las técnicas de resolución de problemas. En segundo lugar, POL (*Project Oriented Learning*) consiste en la realización de un proyecto basado en tareas difíciles. Las principales diferencias entre los dos modelos son: que en PBL el grupo de alumnos debe ser reducido mientras que en POL, el número de alumnos no está determinado. Es más, el rol del profesor en PBL es estimular el razonamiento crítico mientras que en POL es de planificar y consultar.

El modelo PBL (*Problem Based Learning*) es el aprendizaje que resulta del proceso de trabajo hacia la comprensión o la resolución de un problema. El problema se encuentra primero en el proceso de aprendizaje, es decir, en el inicio del proceso de aprendizaje primero se debe presentar a los alumnos el problema y más adelante se les presenta otras aportaciones (clases de teoría o laboratorios) para adquirir el conocimiento (Barrows & Tamblyn, 1980). Los alumnos aprenden en pequeños grupos a través de la participación en un problema real, buscando la información que se necesita para comprender el problema y obtener una solución bajo la supervisión de un tutor. Cada grupo de alumnos investiga, desarrolla una hipótesis y trabaja individual y grupalmente en la solución del problema. El profesor supervisor y juez que gestiona el proceso de aprendizaje ayuda a resolver conflictos. Además guía el aprendizaje a través de preguntas, sugerencias y aclaraciones (Ellis, et al., 1998).

El modelo POL (*Project Oriented Learning*) es un método sistemático de enseñanza que involucra a los alumnos en el conocimiento y las habilidades de aprendizaje a través de un proyecto. Se enfoca en un problema o proyecto que debe ser resuelto mediante el siguiente proceso llevado por los alumnos: la descripción del contexto del proyecto, la búsqueda bibliográfica, el análisis, la valoración de las posibles alternativas, el diseño y la elaboración del proyecto y finalmente la autoevaluación del aprendizaje obtenido. El profesor actúa como experto, tutor y evaluador mientras el alumno protagonista actúa como diseñador, gestor de recursos de aprendizaje y del tiempo. En este modelo, los alumnos investigan más, piensan, toman decisiones y resuelven problemas (Blumenfeld, et al., 1993).

2.4.2 Aprendizaje Colaborativo Soportado por Computador: CSCL

Trabajar en grupo es siempre más enriquecedor y puede llevar a resultados mejores. Según Dillenbourg (1999) el aprendizaje colaborativo “describe una situación en la que se espera que ocurran ciertas formas de interacción entre personas, susceptibles de promover mecanismos de aprendizaje, sin ninguna garantía de que las interacciones esperadas vayan a ocurrir”. Según el enfoque CSCL (*Computer Supported Collaborative Learning*), las nuevas herramientas de aprendizaje deben apoyar el aprendizaje colaborativo con la ayuda de un ordenador, proporcionando la comunicación y el intercambio de trabajo entre los alumnos. El aprendizaje colaborativo se puede ver como el conjunto de estrategias de enseñanza que dependen de la interacción de un pequeño grupo de aprendices, percibiéndose esta interacción como característica central de las tareas de aprendizaje en la clase (Dodl, 1990).

El aprendizaje colaborativo soportado por Computador (CSCL) se basa en tres grandes corrientes (Koschmann, 1996): la escuela del socio constructivismo, las teorías socioculturales y el enfoque situacional del aprendizaje.

En primer lugar, según la teoría Neo-Piagetian (Knight & Sutton, 2004) del constructivismo social, el aumento del desarrollo cognitivo es el resultado de la interacción y el compartir con otros la visión de la realidad y el medio ambiente. Un aspecto importante de la teoría Socio-Constructivista son los conflictos Socio-Cognitivos (Mugny & Doise, 1978) que se producen entre las personas en un ambiente de aprendizaje colaborativo a través de la resolución de la contradicción de ideas entre alumnos, así los alumnos notan la relatividad y la debilidad de sus conceptos y adquieren habilidades

de comunicación. Estos conflictos Socio-Cognitivos producen un aumento en el desarrollo cognitivo individual y un aumento del progreso cognitivo individual en la inteligencia y la comunicación.

En segundo lugar, la teoría Socio Cultural de Vygotsky (Palincsar, 1998) ha señalado las influencias culturales y sociales en el desarrollo cognitivo y describió el desarrollo cognitivo como un proceso de colaboración. La teoría de Vygotsky reconoce que existe una naturaleza social fundamental para el aprendizaje dando lugar a la utilización de la colaboración como una técnica de aprendizaje. Vygotsky cree que el desarrollo individual tiene sus orígenes en fuentes sociales. Además sugiere que la experiencia que sucede a través de la interacción social con otros individuos hace surgir a posteriori nuestras propias reflexiones mentales. Vygotsky define la de zona de desarrollo próximo como la distancia entre el nivel real de desarrollo determinado por la capacidad independiente de un persona y el nivel de desarrollo potencial determinado a través de la ayuda o la colaboración de un adulto

En tercer lugar, el enfoque situacional o la teoría contingencia es un enfoque nuevo que responde a las limitaciones que presentan los enfoques conductismo y cognitivismo, tomando en cuenta el contexto social y el aprendizaje en interacción con otros. La teoría situacional enfatiza el aprendizaje como algo relacionado con la situación, con la cognición individual y el significado de ser construido social y culturalmente. Esto implica trabajar en colaboración, abordando problemas auténticos contextualizados y la negociación del significado a través de la práctica. Existen dos visiones del aprendizaje situado. La primera, derivado del marco psicológico para el diseño de ambientes de aprendizaje y la segunda, respecto al marco antropológico. La perspectiva psicológica destaca la cognición y el aprendizaje a través de actividades situadas en situaciones cotidianas reales y la construcción de oportunidades para actividades auténticas de aprendizaje para el propósito de la transmisión y la adquisición de significado. Mientras que la visión antropológica se centra en el aprendizaje dentro de comunidades de prácticas reales, identidades e interacciones, así como los significados, se construyen dentro de este entorno de manera que el desarrollo integral de la persona se produce y no existe un límite entre el individuo y el mundo (Barab & Duffy, 2000).

El enfoque CSCL se caracteriza por la estimulación del pensamiento crítico y el desarrollo de habilidades de pensamiento de alto nivel (Gokhale, 1995) mediante la búsqueda, la crítica de evidencias y el intercambio de argumentos y opiniones entre los alumnos. Además, este tipo de

aprendizaje puede hacer que las clases sean más agradables para los alumnos y ayude a aclarar las ideas a través de la discusión y el debate mejorando la motivación del aprendizaje. Es importante saber que el uso del enfoque CSCL mejora la calidad del aprendizaje, mejorando la actitud del alumno, su rendimiento académico y su retención (Prince, 2004). También crea un ambiente personal en clases grandes y anima a los alumnos a trabajar juntos, a compartir sus conocimientos y desarrollar sus habilidades, por lo que, se logran los objetivos de aprendizaje (Inaba & Mizoguchi, 2004). CSCL permite la comunicación entre los alumnos y proporciona una mayor cercanía entre los miembros del grupo, de esta manera, pueden lograr su objetivo por medios indirectos. En este enfoque, el nuevo rol del profesor es motivar y guiar al alumno, mientras que los alumnos adquieren más autonomía.

La consecución de los beneficios de aprendizaje colaborativo no siempre depende solamente de la interacción entre los alumnos sino de cómo interactúan los alumnos entre sí y de qué papel juegan en el aprendizaje colaborativo. Es importante asignar un papel apropiado para cada alumno. Los beneficios educativos dependen de qué funciones desempeña el alumno en el aprendizaje colaborativo (Inaba & Mizoguchi, 2004). Es más, el profesor se enfrenta a muchas dificultades durante el uso de este enfoque, tales como la dificultad para organizar experiencias de aprendizaje activo (Michael, 2006).

Por otro lado, es un desafío valorar el trabajo en grupo "*peer evaluation*" de una manera equitativa por las contribuciones de cada miembro del grupo a las tareas de colaboración. Sin embargo, según Chinn (2005) la evaluación por pares "*peer evaluation*" tiene el potencial de fomentar y mejorar las habilidades de pensamiento crítico, de tal manera que los alumnos son capaces de aplicar sus habilidades críticas a problemas más complicados a medida que adquieren más conocimientos, llegando así al nivel más alto de la taxonomía de Bloom, el nivel de evaluación. Papalaskari (2003) también enumera varias ventajas de utilizar la evaluación por pares como por ejemplo que los alumnos ven los errores producidos por otros ya que es difícil reconocer los suyos y estarán obligados a explicar mejor su pensamiento acerca de la respuesta correcta.

Según Fisker y sus colegas (2008), los aspectos que debe tener una herramienta CSCL para soportar las tareas de trabajo en grupo son:

1. Intercambio de artefactos. El intercambio de los artefactos en construcción. En el caso de desarrollo de software puede ser: código fuente, documentos de diseño, diagramas y otros archivos de proyecto.
2. Comunicación. La comunicación es un aspecto importante del trabajo de grupo, hay muchas herramientas de comunicación existentes como por ejemplo mensajería instantánea, correo electrónico, SMS, sitios web de redes sociales y videoconferencias.
3. *Awareness*. Se refiere a la conciencia del grupo (la información que nos hace sentir que estamos en un grupo). La disponibilidad de esta información en el momento oportuno y de forma relevante permite a los miembros del equipo coordinar sus actividades. Esta información, se puede dividir en dos categorías: información generada implícitamente (información que se puede generar como efecto secundario de las acciones comunes de los desarrolladores), e información generada explícitamente (información añadida con el único propósito de informar a los miembros del equipo a través de acciones adicionales).

2.4.3 Sistemas CSCL para la Enseñanza de Algoritmos/Programación

Con respecto a los sistemas CSCL para el aprendizaje de algoritmos, podemos citar la herramienta **CAROUSEL** (Hübscher-Younger & Narayanan, 2003) que se construyó para facilitar el intercambio de animaciones creadas, las evaluaciones y la discusión entre los alumnos, de tal manera, que cada alumno crea una representación en un tiempo determinado y la comparte con el resto de alumnos. Cada actividad cubre un algoritmo determinado y consiste en que los alumnos crean y comparten la representación de ese algoritmo, evaluando y comentando las representaciones de sus compañeros con la clase. La eficiencia educativa de esta herramienta ha sido evaluada positivamente, es decir, mejora el aprendizaje. El sistema permite a los alumnos evaluar estas representaciones basándose en seis características utilizando una escala Likert de cinco puntos: utilidad (¿lo central e importante qué es esta representación para la comprensión del algoritmo?), comprensibilidad (¿qué fácil es esta representación de entender?), relevancia (¿cómo de bien señala esta representación las características importantes del algoritmo?), la familiaridad (¿cómo de familiarizado estabas con el contenido de la representación?), el placer (¿cuánto te gustó la forma de la explicación del algoritmo en esta representación?), originalidad (¿cuánto se diferencia esta representación de la otras representaciones?)

Disviz (Sherstov, 2003) es también un sistema de visualización que proporciona la animación de una muestra representativa de algoritmos de grafos. Ha sido desarrollado para permitir el aprendizaje en grupos a través de redes locales en los que se pueden crear animaciones de algoritmos de grafos que pueden ser vistos por todos los miembros de la red. Cada solicitud de ejecución del algoritmo está dada por el nombre de host de su emisor y el momento de la emisión. Disviz proporciona la animación de los siguientes algoritmos fundamentales de grafos: Dijkstra's, búsqueda en amplitud, Barůvka para el árbol de recubrimiento de coste mínimo.

Existen numerosas herramientas basadas en el enfoque CSCL para el aprendizaje de programación a través de la visualización o animación de programas. Por ejemplo, la herramienta **JeCo** (Nuutinen, 2006) es una combinación de una herramienta de visualización para programas Java, llamada **Jeliot3** (es una aplicación de visualización de programas que hace uso de la animación para mostrar las llamadas a métodos, variables de operación permitiendo al alumno seguir paso a paso la ejecución de un programa generando automáticamente toda la visualización) y una herramienta de edición colaborativa **Woven Stories** (Gerdt, et al., 2001) para la escritura colaborativa.

Otra herramienta colaborativa descrita por Chang & Chen (2008) para el aprendizaje del lenguaje C ofrece comunicaciones de voz y *chat* para apoyar el aprendizaje colaborativo de tal manera que un grupo de alumnos puede estudiar, diseñar, encontrar y resolver un problema en el lenguaje de programación C a través de la discusión.

SICAS-COL (Santos, et al., 2010) es otra herramienta que visualiza el código fuente haciendo animación del mismo mediante diagramas de flujo, visualizando gráficamente el comportamiento de las expresiones, estructuras de datos y pila de ejecución, para el uso de un grupo de alumnos. Este sistema permite a los alumnos implementar algoritmos para resolver problemas utilizando una representación de diagramas de flujo, incluso ejecutar y simular la solución paso a paso continuo a una velocidad baja o rápida. **H-SICAS** (Marcelino, et al., 2008) es la adaptación de SICAS para su utilización en dispositivos móviles.

Otros sistemas, como **AURA** (Kumar, et al., 2010) tienen el objetivo de aprender habilidades para desarrollar algoritmos y ponerlas en práctica en varios lenguajes de programación. AURA es un entorno innovador para la generación de código que permite a los alumnos aprender los conceptos

básicos de los lenguajes de programación mediante métodos de arrastrar y soltar (*drag-n-drop*). Después se crea el código, se genera el pseudocódigo y se muestra en la ventana de script. Esto da a los alumnos una visión general de una implementación. También goza de facilidades de colaboración como compartir el trabajo entre los alumnos y permitir a los profesores revisar/editar el trabajo de los alumnos desde cualquier lugar.

Otros trabajos proponen herramientas para la programación pura como es el caso del sistema **COLLEGE** (*COLLaborative Edition, compilinG and Execution of programs*) (Amrit, et al., 2007) que soporta compilación y ejecución facilitando el aprendizaje colaborativo de la programación. Se compone principalmente de una herramienta de mensajería instantánea (*chat*) y una herramienta de toma de decisiones. Además, el sistema ofrece funciones de *awareness* para facilitar la percepción y la realización de trabajos en grupo.

El sistema **TurtleGraph** (Chang & Jehng, 1998) también es otro entorno de colaboración en la programación de LISP-LOGO para la enseñanza de la programación recursiva, donde el alumno realiza tareas colaborativas de aprendizaje visual, como escribir en colaboración programas de LISP-LOGO para resolver problemas de dibujo de patrones geométricos que representan explícitamente la estructura profunda de los procedimientos recursivos. El sistema tiene la capacidad de hacer el seguimiento de las conversaciones de los alumnos y la visualización de estas conversaciones en la pantalla. Se pide a los alumnos que escriban programas para dibujar un patrón siguiendo una secuencia predefinida de dibujo.

La depuración es una forma de verificar la corrección de un algoritmo. Consiste básicamente en explorar el algoritmo, ejecutándolo paso a paso y comparar en todo momento los valores que van tomando las distintas variables con los valores esperados. El objetivo de la depuración es el de "obtener un programa que funciona con todos los datos que un usuario podría proporcionar". Un depurador permite interactuar e inspeccionar el programa en ejecución, por lo que es posible rastrear el flujo de ejecución y localizar los problemas.

Existen muchas herramientas para la enseñanza de la programación que soportan funcionalidades propias de entornos de desarrollo profesionales incluidas la depuración. Por ejemplo, el sistema **COALA** (Jurado, et al., 2012). Es un entorno para el aprendizaje de algoritmos, está basado en el IDE Eclipse que proporciona un entorno de desarrollo enriquecido para apoyar a

las actividades de los alumnos y de los profesores. Por su parte, el sistema **Cole-Programming** (Jurado, et al., 2012) que extiende el plug-in de COALA con utilidades de interacción colaborativa proporcionando herramientas colaborativas para la programación en parejas (*Pair programming*). Es un sistema distribuido que dota de herramientas colaborativas asíncronas y sincrónicas (*chat*, foros o herramientas de votación), además de funciones de *Awareness* y permite la especificación de actividades de programación por parte del profesor.

Jazz Samgan (Vijay, et al., 2008) también fue creada para apoyar la funcionalidad de programación en parejas distribuidas, basándose en el IDE Eclipse, proporcionando herramientas de gestión de proyectos, mensajes instantáneos y control de versiones. Por otra parte, Fisker y otros (Fisker, et al., 2008) presentan una herramienta integrada en el IDE BlueJ para apoyar el trabajo en grupo. Es un sistema para compartir el trabajo entre alumnos usando BlueJ que emplea CVS para la actualización de trabajos y goza de facilidades de *Awareness*.

Otras herramientas son interfaces basadas en Web para el aprendizaje de programación como es el caso del sistema **Adessowiki** (Rittner, et al., 2011) que se compone de una colección de páginas Web editables que forman una wiki, diseñado para facilitar la creación colaborativa de contenido científico dedicada al procesamiento de imagen y soporte para Python / C / C + +. Se trata de una plataforma de colaboración para la programación científica y la redacción de documentos. **SICODE** (Pérez, et al., 2006) por su parte, es una herramienta Web de desarrollo de software para el aprendizaje del lenguaje Java. Se basa en técnicas de procesadores de lenguaje que permiten realizar un análisis estático del código fuente para descubrir los errores de programación presentes en el código, además goza de funcionalidades de colaboración que permiten programar entre un grupo de alumnos, colaborar en la solución de errores y en la mejora del código fuente.

Finalmente, el sistema **HabiPro** (Vizcano, et al., 2000) es una herramienta pedagógica y colaborativa destinado a desarrollar buenos hábitos de programación para los alumnos principiantes. No es un software para la enseñanza de la programación sino para desarrollar en los alumnos las habilidades de un principiante como la observación, la reflexión o la estructura que son necesarias para llegar a ser un buen programador. El sistema presenta cuatro tipos de ejercicios, encontrar un error en un programa, poner un programa en el orden correcto, predecir el resultado y completar un

programa. Cuando un grupo propone una solución, si no es correcta el sistema proporciona mensajes de ayuda.

2.5 Evaluación de Usabilidad en Sistemas Educativos

En esta sección se recogen todos los métodos de evaluación de usabilidad en general y para sistemas educativos en particular. Primero, se ofrece una definición de usabilidad. Después, se detallan las principales técnicas de evaluación de usabilidad. Finalmente se incluyen algunos trabajos relacionados.

2.5.1 Definición de Usabilidad

El término usabilidad, se puede definir según la Organización Internacional para la Estandarización (ISO), como la medida en que un producto puede ser usado por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en condiciones específicas de uso (Abran, et al., 2003).

Según Nielsen (2012), la usabilidad es un término multidimensional que indica que un sistema usable debe poseer los siguientes atributos:

1. Aprendizaje, ¿Cómo de fácil es para los usuarios cumplir las tareas básicas la primera vez que se encuentran con el diseño?
2. Eficiencia, una vez que los usuarios ha aprendido el diseño, ¿Cómo de rápido desarrollan sus tareas?
3. Memoria, cuando los usuarios regresan al diseño después de un periodo de no usarlo, ¿Cómo de fácil pueden restablecer las destrezas?
4. Errores, ¿Cuántos errores cometen los usuarios?, ¿Cuánto de graves son estos errores? y ¿Cómo de fácil se recobran de los errores?
5. Satisfacción, ¿Qué tan agradable es de usar un determinado diseño?
6. Utilidad, se refiere a la funcionalidad del diseño, ¿Hace lo que el usuario necesita?

Nielsen señala que la aceptabilidad de un sistema es una combinación de su aceptabilidad social y de su aceptabilidad práctica. Lo social tiene que ver con la aceptación que un grupo de personas puede dar a un sistema. Lo práctico incluye costes, soporte, confiabilidad, compatibilidad

con los sistemas existentes, usabilidad y utilidad que implica que el sistema responda al objetivo para lo cual fue creado.

La usabilidad debe aplicarse para el desarrollo de cualquier sistema para evitar que sea inutilizable, para mejorar la coherencia visual, la legibilidad y asegurar la satisfacción del usuario. Es importante poner en práctica la usabilidad en las herramientas educativas debido a varios factores como por ejemplo, evitar el fracaso de la funcionalidad del sistema, es decir, si los usuarios no son capaces de utilizar el sistema no podrán usarlo y entonces el sistema no cumplirá con sus objetivos.

Los beneficios de usabilidad implican una reducción y optimización general de los costes de producción, así como un aumento en la productividad de las herramientas analizadas. La usabilidad permite mayor rapidez en la realización de tareas y reduce las pérdidas de tiempo (Sulaiman, et al., 2009). Por eso, es imprescindible incluir la evaluación de la usabilidad en el ciclo de vida del desarrollo del software, tanto en el desarrollo evolutivo como el incremental (Sayago, et al., 2003) de tal manera que el ciclo de vida reestructurado quedaría como puede verse en la Ilustración 1:

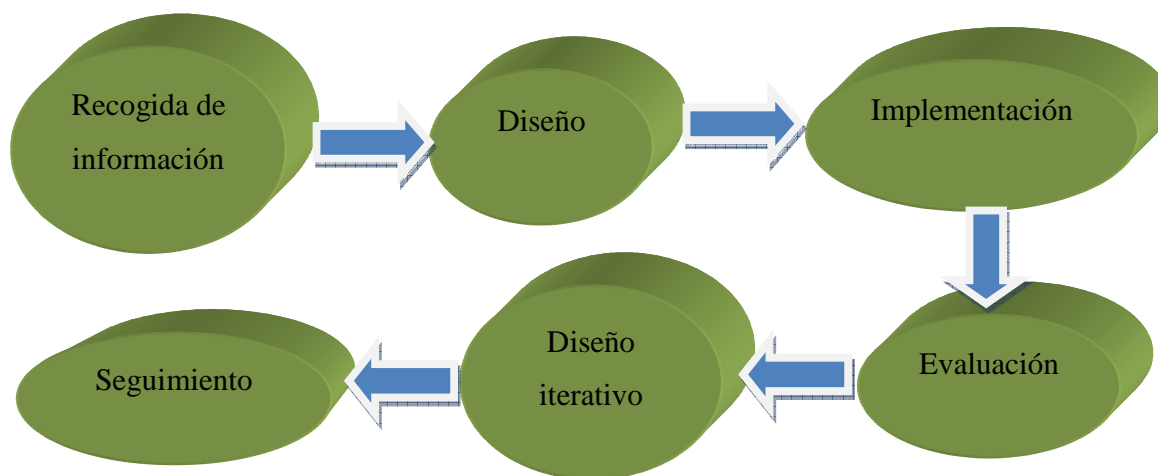


Ilustración 1. Ciclo de vida del software

2.5.2 Principales Técnicas de Evaluación de Usabilidad

Para garantizar la utilización de los sistemas de la mejor manera es importante asegurar que tanto los procesos y los sistemas en desarrollo son evaluados. La evaluación se aplica por múltiples propósitos en relación con las distintas fases y funciones del proceso del desarrollo de un sistema. Diferentes metodologías serán empleados en las diferentes etapas. Los resultados de las evaluaciones de usabilidad se necesitan con rapidez para que el desarrollo no se retrase y puede llevar a cabo las

evaluaciones múltiples de sucesivos ciclos de desarrollo. Existen dos principales técnicas de evaluación que son la inspección de usabilidad y el *test* de usuarios (Riihiaho, 2009) (Blecken & Marx, 2010).

2.5.2.1 Inspección de Usabilidad

El primer grupo de las técnicas de evaluación de usabilidad reúne todos los métodos que tienen que ser llevados a cabo por expertos en usabilidad. Estos métodos son los más adecuados para las evaluaciones en la etapa inicial de la fase de desarrollo del sistema.

1) Evaluación Heurística en un Software Educativo

Una de las técnicas de la inspección de usabilidad y de la ingeniería de la usabilidad es la evaluación heurística que es la menos costosa por su requerimiento de pocos recursos y la disminución del coste asociado a la producción de software.

La evaluación heurística, como su nombre indica, es la evaluación o la verificación de una serie de principios heurísticos. Se considera como un método que evalúa y describe los problemas que se encuentran en una herramienta (Sulaiman, et al., 2009), (Alsumait & Al-Osaimi, 2009), (Nielsen, 1992).

La evaluación heurística o por expertos se encuadra dentro de los métodos de inspección de la usabilidad de un sistema. Se basa en el recorrido y el análisis del sistema. A partir de aquí se localizaran errores y problemas de diseño. El resultado será un informe que dé cuenta de las mejoras necesarias para que el sistema sea totalmente usable (Albion, 1999). Lo que se pretende es evitar repetir los errores en el rediseño de la interfaz, entonces es conveniente jerarquizar la gravedad de estos errores para ir actuando según lo importante que sea. En este sentido habrá que tener en cuenta tres factores:

- La frecuencia con que ocurre el problema.
- El impacto que tiene el problema con los usuarios.
- La persistencia del problema en todo el sistema.

El primer paso es la planificación, que empieza con la selección de los evaluadores que deberán llevar a cabo la etapa siguiente en el proceso de evaluación y tienen que ser entre tres o cinco evaluadores según Nielsen y Landauer (González, et al., 2006). Participarán tres tipos de evaluadores que son: los expertos en usabilidad, los desarrolladores y los usuarios potenciales (son usuarios inexpertos que conocen el sistema a evaluar). El equipo evaluador debe seleccionar los principios heurísticos a evaluar teniendo en cuenta los puntos principales a cubrir en el sistema a evaluar y luego adecuarlos al contexto de uso. Cada principio heurístico es equivalente a una serie de preguntas o sub-principios heurísticos capaces de instanciarse de manera natural en el sistema a evaluar. Se debe elegir una escala de valores tipo (1, 2, 3,4) para contestar estas preguntas (Benson, et al., 2002) (para que nos ayude a mejorar el análisis de los resultados):

- [1=] Problema sin importancia: no necesita arreglarse a menos que haya tiempo de sobra.
- [2=] Problema de poca importancia: arreglarlo no tiene mucha importancia.
- [3=] Problema grave: es importante arreglarlo.
- [4=] Catástrofe: es obligatorio arreglarlo.

Sin embargo algunas preguntas deberán ser respondidas sólo usando el lenguaje natural entre el equipo evaluador de la herramienta. Finalmente podemos llegar a formular una plantilla donde hay un espacio para escribir las heurísticas que son reglas generales que describen las propiedades comunes de las interfaces usables (por ejemplo, la visibilidad del estado del sistema, que se refiere a mantener los usuarios informados sobre lo que está pasando a través de información adecuada en un plazo razonable) y que luego se medirán indicando la escala de valores, el significado de cada valor, un espacio para la respuesta de estas heurísticas y finalmente un espacio para las observaciones de los evaluadores.

Una vez terminada la fase de planificación, los evaluadores deben proceder a la inspección de la interfaz del sistema a evaluar de manera individual para asegurar evaluaciones independientes e imparciales de cada evaluado y luego podrán comunicar sus hallazgos. Según Nielsen, es recomendable tener sesiones de evaluación de una o dos horas por cada parte de la interfaz a evaluar, usando la lista de comprobación generada y añadiendo principios heurísticos.

Los tipos comunes de la evaluación heurística aplicada en muchos sistemas son: las diez heurísticas de Jacob Nielsen (2012) que es ampliamente utilizado para todos los sistemas y que Squire y Preece (1996) las han adaptado para las aplicaciones educativas; las Ocho reglas de oro de Schneiderman (González, et al., 2006), que es más conocido para las interfaces interactivas; y las heurísticas de la evaluación de usabilidad de Reeves (Benson, et al., 2002), que es la más adecuada para ser aplicada a los sistemas educativos incluidas en la Tabla 3.

Tabla 3. Las heurísticas de Reeves

<i>PRINCIPIO HEURÍSTICO</i>	<i>DESCRIPCIÓN</i>
Interactividad	La herramienta debe ofrecer las interacciones relacionadas con el contenido y las tareas que apoyan el aprendizaje significativo.
Mensaje de Diseño	La herramienta debe presentar información de acuerdo con principios de procesamiento de información.
Diseño de aprendizaje	Las interacciones en la herramienta han de ser diseñados de acuerdo con sólidos principios de la teoría del aprendizaje.
Integración de los medios de comunicación	La inclusión de los medios de comunicación en la herramienta debe ofrecer claros fines pedagógicos y/o motivación.
Instrucciones de evaluación	La herramienta debe ofrecer oportunidades de evaluación que estén alineados con los objetivos del programa y el contenido.
Recursos	La herramienta accede a todos los recursos necesarios para apoyar el aprendizaje efectivo.
Comentarios	La herramienta proporciona información que es contextual y relevante para el problema o la tarea que lleva a cabo el alumno.

En resumen, el desarrollo de las herramientas educativas requiere una cantidad inmensa de tiempo y dedicación para obtener un sistema de uso docente usable. Por eso, el uso de la usabilidad es muy importante, no solo porque nos ayuda a medir la facilidad de uso de los sistemas, sino nos ayuda a medir la facilidad de aprendizaje en el caso de las herramientas educativas.

2) Recorrido Cognitivo (*Cognitive Walkthrough*)

Este método es una técnica de revisión donde los evaluadores expertos construyen escenarios para las tareas a partir de una especificación o de un prototipo temprano para desempeñar después el papel del usuario trabajando con la interfaz en cuestión. Se trata de reconstruir paso a paso la interacción del usuario con el sistema (Wharton, et al., 1994).

El método asume que el experto aprende a usar la interfaz de usuario comparando lo que pensaba hacer con lo que realmente hizo. En el recorrido cognitivo se le pide al experto que vaya realizando cada una de las acciones señaladas, de acuerdo a la secuencia dada en el escenario de tareas. El experto interactúa con el sistema anotando y contrastando con la secuencia de acciones predeterminadas y contestando a las siguientes cuatro preguntas sobre las expectativas de comportamiento de los usuarios:

1. ¿El usuario puede lograr el efecto deseado?
2. ¿El usuario puede observar que la acción correcta está disponible?
3. ¿El usuario puede asociar la acción correcta con el efecto a conseguir?
4. Si se realiza la acción correcta, ¿el usuario puede ver que se está avanzando hacia la solución de la tarea?

3) GOMS (Método de modelos)

Los modelos GOMS (*Goal, Operations, Methods and Selection Rules*) son una descripción del conocimiento que un usuario debe tener para poder llevar a cabo una tarea sobre un dispositivo o un sistema. Es un análisis cuantitativo de las acciones para predecir el tiempo requerido por un usuario para completar las tareas. Un análisis o modelo GOMS se constituye a base de describir las Metas, Operadores, Métodos y Reglas de Selección para un conjunto de tareas de una manera formal. Las metas son los objetivos de los usuarios, los operadores son acciones que se realizan para llegar a la meta, los métodos son secuencias de los operadores para cumplir una meta y finalmente las reglas de selección son la descripción de la selección de un determinado método sobre los demás (John & Kieras, 1994).

2.5.2.2 Pruebas de Usuarios

Las pruebas (*test*) de usuarios son un tipo de evaluación complementaria a la inspección de usabilidad y se basa en la observación y el análisis de la utilización de la interfaz del usuario y se considera la manera más cercana de aproximarse al uso real de una interfaz. Si llevamos a cabo un *test* de usuarios sin haber realizado previamente una evaluación heurística, probablemente prestaremos demasiada atención a problemas de uso que podrían haber sido descubiertos con una simple evaluación heurística.

Como toda evaluación de usabilidad, cuanto más esperamos para su realización más costosa resultará la reparación de los errores de diseño. Un *test* de usuarios es más costoso que una evaluación heurística, por lo que sería desperdiciar tiempo y dinero el utilizarlo para descubrir errores de diseño motivados por el incumplimiento en el diseño de principios generales de usabilidad (heurísticos). En este tipo de evaluación los usuarios finales interactúan con el sistema en sus diferentes etapas de desarrollo. La evaluación con usuarios es importante debido a que:

1. Los usuarios tienen su propia forma de pensar diferente a los diseñadores de sistema.
2. Los usuarios finales tienen un perfil perfecto relacionado con el sistema a desarrollar.
3. Los usuarios que realizan la evaluación actuarán de manera diferente, es decir, elegirán caminos o rutas diferentes de las que usarán comúnmente los desarrolladores del sistema que para los desarrolladores del sistema sería lo “común”.
4. Ningún sistema se desarrollará de manera correcta sin conocer el usuario final.

El proceso del *test* de usabilidad con usuarios según Hansen (Riihiho, 2009) consiste en 10 etapas que son:

1. Obtener una información a fondo sobre la evaluación de la usabilidad.
2. Hacer un plan de pruebas.
3. Diseñar el *test*.
4. Organizar un entorno para el *test*.
5. Llevar a cabo un *test* piloto.
6. Reunir a los participantes.
7. Establecer la sala de ensayo.
8. Llevar a cabo el *test*.
9. Compilar y analizar los resultados.
10. Recomendar cambios.

1) Métodos de recogida de datos para pruebas de usabilidad

A continuación, se enumeran varios métodos de recogida de datos más comunes (Lee, 1999).

1. **Entrevistas:** es una técnica común en las evaluaciones de usabilidad, donde el usuario debe responder a una serie de preguntas comenzando con los aspectos de alto nivel y luego de bajo nivel, tales como su experiencia con el sistema, actitud e impresiones. Esta técnica permite al usuario expresar sus pensamientos.
2. **Cuestionarios:** es un método para la obtención, registro y la recolección de información. Es un formulario escrito, estructurado y completado por el usuario. Se considera como un procedimiento menos flexible que una entrevista pero permite llegar a más usuarios y generar informes estadísticos con mayor facilidad obteniendo datos cualitativos y cuantitativos.
3. **Pensamiento en voz alta (*Thinking aloud*):** es un método que ayuda en la captura de un amplio rango de actividades cognitivas, de tal forma que durante el transcurso del *test*, cuando el usuario está realizando una tarea, se le pide que exprese en voz alta sus pensamientos, sensaciones y opiniones mientras interactúa con el sistema.
4. **Co-descubrimiento:** es una derivación del método pensamiento en voz alta (*Thinking aloud*) que implica tener dos usuarios en vez de uno. La principal ventaja es que resulta mucho más natural que el método de pensar en voz alta, ya que las personas normalmente verbalizan más cuando tratan de resolver un problema conjuntamente y además hacen muchos más comentarios; la desventaja es que los usuarios pueden tener diferentes estrategias de aprendizaje.
5. **Grupo de discusión dirigida (*focus group*):** los grupos de discusión dirigida son los más utilizados como una técnica de evaluación. Se reúnen 4 a 9 usuarios para discutir aspectos relacionados con el sistema. El evaluador prepara la lista de aspectos a discutir y recoge la información que necesita de la discusión. Este método tiene como objetivo permitir capturar reacciones espontáneas del usuario e ideas que evolucionan en el proceso dinámico del grupo. Los participantes son motivados a dar libremente su opinión honesta sobre el sistema, incluyendo sugerencias para mejorarlo.
6. **Soporte informático.** Dos métodos para reunir información detallada sobre el usuario final son el uso de videos para la captura de las interacciones del usuario durante las pruebas de usabilidad y la recogida de datos del sistema a través del auto-registro que se utiliza para rastrear las acciones del usuario (logs).
7. **Observaciones.** Las observaciones son un método complementario a las técnicas de evaluación de usabilidad. Según Cohen y sus colegas (Cohen, et al., 2007), la observación puede ser una herramienta de investigación muy útil para hacerse una idea de las situaciones. Sin embargo,

puede ser costosa en tiempo y esfuerzo y es propensa a las dificultades de interpretación o deducción del significado de los datos. Por otra parte, el principal ventaja de la observación es usar la cognición directa que tiene el potencial de producir datos válidos o más auténticos que de otro modo de evaluación. Además, el rasgo distintivo de la observación como un proceso de investigación es que ofrece al investigador la oportunidad de recopilar datos "en vivo" de forma natural de lo que está ocurriendo en situaciones sociales (Cohen, et al., 2007). Debdí y otros (Debdí, et al., 2012) describen dos evaluaciones de usabilidad de una herramienta educativa mediante la recogida de observaciones. Los resultados de esta evaluación fueron útiles en dos sentidos: 1) Reforzar los hallazgos resultantes de aplicar el método de evaluación de usabilidad por cuestionarios y permitir la identificación de nuevos datos de interés para mejorar el enfoque del aprendizaje usado. Y 2) Aprender nuevas lecciones metodológicas sobre el uso de observaciones como método de investigación.

2) Método Etnográfico (*Contextual Walkthrough*)

Lo que se trata en este método, es hacer un modelo mental del usuario con respecto a la tarea y artefacto. Consiste en seleccionar perfiles de usuarios reales, diseñar entrevistas y organizar sesiones de observación directa del entorno de trabajo, revelando el significado de las acciones y las interacciones de los usuarios finales mediante la observación y las entrevistas que revelan el porqué de las acciones y las decisiones del usuario. Este método se basa en tres principios fundamentales:

1. Entender el contexto en el que un sistema es utilizado.
2. El usuario es parte del proceso de diseño.
3. El proceso de diseño para la usabilidad debe tener un objetivo.

El objetivo de las observaciones en este método es revelar problemas de usabilidad en el sistema de evaluación en lugar de recoger ideas para nuevos proyectos de desarrollo (Riihiahho, 2009).

3) Recorrido Pluralista (*Pluralistic Walkthrough*)

Se trata de reuniones de desarrolladores y expertos con usuarios donde discuten y evalúan cada elemento de la interacción. Ejecutan ciertas tareas sobre el sistema mientras anotan cada paso

de la interacción. Luego se debaten los problemas y decisiones tomadas durante el proceso. Combina tres visiones diferentes del sistema: lo que se debe (expertos de usabilidad), se quiere (los usuarios) y se puede (desarrollador). Con este tipo de grupos se cuenta con la ventaja de un diverso rango de capacidades y perspectivas para resolver los problemas de usabilidad. La desventaja que es difícil de encontrar un contexto adecuado de las tareas a realizar para las pruebas de usabilidad. Las ventajas es fácil de usar, permite realizar pruebas iterativas y cumple con los criterios de todas las partes involucradas en la prueba (Lee, 1999).

3) Recorrido Informal (*Informal Walkthroughs*)

Es una mezcla de tres pruebas de usabilidad: pensar en voz alta, observaciones y entrevistas. En un recorrido informal, no hay tareas de pruebas específicas y predefinidas. En su lugar, el moderador de prueba tiene una lista de características que los usuarios deben seguir a través de su propio ritmo y orden (Lee, 1999). En las sesiones de recorrido informal, se les pide a los usuarios que exploren el sistema en la forma en que harían si estuvieran solos y pensar en voz alta mientras se explora el sistema. Ellos son también animados a comentar sobre el sistema en cualquier momento. El moderador de la prueba podrá interrumpir los usuarios para las preguntas, pero sobre todo les observa.

2.5.3 Experiencias de Evaluación de Usabilidad de Sistemas Educativos

En este apartado de revisan experiencias de evaluación de usabilidad en varios tipos de sistemas educativos con cierta relación con GreedEx y GreedExCol. Sucesivamente se revisan sistemas de visualización, sistemas de corrección automática y sistemas colaborativos para aprendizaje de la programación.

Son diversos los sistemas de visualización de algoritmos que han sido sometidas a una evaluación de usabilidad. Urquiza & Velázquez (2009) contienen una revisión de diversas evaluaciones positivas de sistemas de visualización, incluyendo las evaluaciones de usabilidad. El sistema **ALVIE** (Crescenzi & Nocentini, 2007) fue evaluado satisfactoriamente con un cuestionario basado en cinco preguntas sobre el sistema. El objetivo era evaluar la cantidad de alumnos que apreciaron la integración de una taxonomía de aprendizaje en el curso. Los resultados de la encuesta fueron muy favorables.

Por otro lado, los resultados de la evaluación de usabilidad del sistema **AI-Search** (Ciesielski & McDonald, 2001) indicaron que los alumnos creen que el uso del sistema aumenta su comprensión de los algoritmos de búsqueda.

El sistema de visualización **TRAKLA2** (Malmi, et al., 2004) fue también evaluado positivamente. A los alumnos les gustó mucho el sistema y lo consideraron útil para el aprendizaje (el 94% de los alumnos creen que el sistema es muy fácil de usar). Por otra parte, Lui y sus colegas (Liu, et al., 2007) evaluaron dos paquetes desarrollados en Matlab para el aprendizaje de conceptos matemáticos abstractos y algoritmos de optimización y sus aplicaciones en la resolución de problemas de diseño óptimo. Estos paquetes han sido evaluados mediante un cuestionario con 9 preguntas sobre la usabilidad del paquete y también mediante entrevistas a los alumnos de manera informal. Estas dos evaluaciones mostraron que el paquete de visualización ayuda a comprender conceptos matemáticos abstractos y provee a los alumnos de herramientas útiles para la solución de problemas de diseño óptimo y ayuda a explicar algoritmos matemáticos y procedimientos de diseño óptimo. Sin embargo hubo necesidad de una interfaz de usuario amigable.

El sistema **SRec** también (Pérez-Carrasco, et al., 2010) ha sido desarrollado y evaluado de forma similar a la descrita en esta tesis para GreedEx. SRec ha sido evaluado cinco veces con cuestionarios con el fin de garantizar la utilidad y usabilidad de la herramienta.

El sistema de visualización de analizadores sintácticos **VAST** (Martínez, et al., 2009) fue evaluado con dos métodos: la evaluación de experto y la evaluación observacional. Durante las dos sesiones de observación, se recogieron opiniones de los alumnos donde indicaron la facilidad de uso de VAST pero también detectaron algunos errores en la herramienta.

También se han realizado evaluaciones de usabilidad de varios sistemas de visualización de programas. La herramienta **Ville** (Kaila, et al., 2009) ha sido evaluada mediante un cuestionario estructurado en tres partes: preguntas generales acerca de la herramienta, la utilidad del sistema a la hora de aprender nuevos conceptos de programación y opiniones acerca de las características de la herramienta. El resultado de la encuesta fue que la herramienta es muy adecuada para la enseñanza de la programación y muy útil para la enseñanza de todos los conceptos básicos. Las características de Ville más útiles fueron la visualización de estados variables y la evaluación automática de los ejercicios. En conclusión, los alumnos prefieren integrar los ejercicios Ville con los métodos más tradicionales de la enseñanza.

Algunos sistemas de corrección automática también se han sometido a evaluaciones de usabilidad. Por ejemplo, el sistema **Mooshak** se evaluó pidiendo a los alumnos que indicaran un grado de acuerdo/desacuerdo con una serie de declaraciones, como si aprenden mejor con la nueva metodología, si el sistema fomenta la competitividad, o si la herramienta es muy útil y fácil de usar. Los resultados han sido positivos. Otro sistema de calificación automática que también fue evaluado positivamente es **CourseMarker**. De las preguntas que tenían que valorar los alumnos de una escala del 1 a 5 (donde 1 es muy positiva y 5 es muy negativo) eran sobre la experiencia global con la herramienta, la interfaz, facilidad y utilidad del sistema, etc. Los resultados del cuestionario mostraron que todos los alumnos prefieren CourseMarker, sobre todo por su facilidad de uso, por sus ejercicios que estaban bien diseñados, equilibrados y adecuados para las necesidades de la clase.

Con respecto a los sistemas CSCL que han sido evaluados, podemos citar el sistema **COALA** (Jurado, et al., 2012) que fue sometida a una evaluación de opinión para evaluar la impresión de los alumnos sobre dicha herramienta. El cuestionario presentado constaba de preguntas tipo test y de respuesta abierta. El objetivo de esta evaluación era valorar las herramientas colaborativas que implementa **COLE-Programming** (Jurado, et al., 2012) para evolucionar el sistema COALA, y por el otro lado, valorar las herramientas colaborativas externas no específicas para tareas de programación, comparando ambas aproximaciones. Los resultados han sido positivos dado que los alumnos encontraron atractivas las herramientas colaborativas del sistema en cuestión.

El sistema **Adessowiki** (Rittner, et al., 2011) también fue evaluado positivamente por la calidad de sus cajas de herramientas y su contribución al aprendizaje. Un cuestionario con 16 preguntas fue desarrollado para investigar las opiniones de los alumnos sobre el sistema. La encuesta llevada a cabo confirmó las principales ventajas de la programación en colaboración y la plataforma de la escritura de documentación.

El sistema integrada en el **IDE BlueJ** para apoyar el trabajo en grupo (Fisker, et al., 2008) también fue sometido a una evaluación de usabilidad. Se pidió a los alumnos evaluar en una escala de cinco puntos (desde "muy en desacuerdo" a "totalmente de acuerdo") dos afirmaciones sobre la facilidad de uso y utilidad. Los alumnos evaluaron la herramienta positivamente y les consideraron útil y fácil de usar.

2.6 Evaluación de Motivación de Sistemas Educativos

En esta sección se enumeran los métodos de evaluación de motivación para los sistemas educativos. Asimismo los trabajos realizados hasta ahora en el dominio de la enseñanza.

2.6.1 Motivación y sus Dimensiones

La teoría de la autodeterminación es una teoría contemporánea sobre la base de la motivación humana fundamental que implica la autonomía del alumno y el sentido de la libertad de elección (Martín-Albo, et al., 2009). Existen diferentes tipos de motivación que difieren en sus niveles inherentes de autodeterminación: motivación intrínseca, motivación extrínseca y desmotivación. En primer lugar, la motivación intrínseca se refiere a la motivación que viene desde el interior de una persona sólo por el placer, el deseo o interés en la tarea en sí. Estas personas no necesitan ninguna recompensa o incentivos para hacer algo (Deci & Ryan, 1985).

En segundo lugar, la motivación extrínseca se refiere a la realización de actividades debido a los incentivos o ventajas de la propia actividad. Los tipos de la motivación extrínseca son la regulación externa y la regulación identificada que varían en el grado de autodeterminación porque el comportamiento es iniciado por elección: el que presenta menor autodeterminación se denomina regulación externa y ocurre cuando un individuo se involucra en una actividad para obtener la recompensa o para evitar el castigo, sin embargo, la regulación identificada ocurre cuando la conducta se considera importante porque la actividad es de mérito, es decir, el sujeto condiciona su conducta porque él cree que los demás lo ven importante o porque él mismo cree que es importante y se ve en la obligación, simplemente reconoce que un comportamiento es beneficioso hacia su desarrollo, por ejemplo, realizar ejercicios diarios porque son parte de un objetivo general para mejorar el rendimiento.

En tercer lugar, el concepto que presenta menos autodeterminación es la desmotivación, que se produce cuando las personas no pueden percibir las contingencias entre el comportamiento y los resultados, sus comportamientos no son ni la motivación intrínseca ni extrínseca. Es decir, la persona no realiza una actividad porque no la valora o no espera un resultado positivo de la misma. No relaciona el comportamiento con el resultado esperado.

Según la teoría de la autodeterminación (Deci & Ryan, 1985), estas dimensiones de motivación se ordenan a lo largo de la autodeterminación, es decir, en un extremo está la desmotivación y en el otro extremo está la motivación intrínseca. Estos tipos de motivación están ordenados gradualmente en sus niveles de autodeterminación, donde los niveles más altos están correlacionados con la motivación intrínseca y la regulación identificada y los niveles más bajos están vinculados a la desmotivación y la regulación externa. Estas dimensiones de motivación son valoradas por los individuos, lo cual están asociadas con efectos positivos la motivación intrínseca y la regulación identificada y con efectos negativos la desmotivación y la regulación externa.

2.6.2 Métodos de Evaluación de Motivación

La motivación situacional es la motivación de los individuos mientras se están involucrando en una actividad. El modelo jerárquico de motivación intrínseca y extrínseca de Vallerand (1997) ofrece una perspectiva multinivel sobre la motivación humana. Se distinguen tres niveles de análisis de los tipos de motivación (intrínseca, extrínseca y desmotivación): el global, el contextual y el situacional. El nivel global se refiere a una orientación motivacional general de interacción con el entorno de una manera intrínseca, extrínseca o desmotivacional. El nivel contextual es una orientación motivacional individual hacia un contexto específico o un conjunto de actividades específicas relacionadas. El nivel situacional se refiere a las experiencias motivacionales individuales cuando se participa en una actividad específica en un momento concreto.

Existen varios instrumentos para medir la motivación situacional. Por ejemplo, la Escala de Motivación Situacional (AMS) (Martín-Albo, et al., 2009) es un instrumento para evaluar la motivación situacional en el entorno educativo. Cuenta con 14 ítems y evalúa las dimensiones de la motivación intrínseca, la regulación identificada, externa y desmotivación. Todos los ítems responden a la pregunta “¿Por qué crees que deberías realizar la actividad que has hecho?”. Los ítems están medidos bajo una escala de 1 (no se corresponde en absoluto con lo que pienso) al 7 (se corresponde exactamente con lo que pienso).

Otro instrumento es la Escala de Orientación de Motivación Intrínseca vs. Extrínseca descrita por Harter (1981). Es un instrumento adecuado para medir la orientación motivacional de los alumnos en el aula y cuenta con 30 ítems con 5 subescalas: (1) *Preferencia por el reto vs. Preferencia por el trabajo fácil*, (2) *Curiosidad e interés vs. Agradar al profesor, obtención de notas*

(3) *Tendencia al dominio y competencia independiente vs. Dependencia del profesor*) (4) *Juicio independiente de las actividades vs. Juicio dependiente del profesor* y (5) *Criterio interno de evaluación de las tareas vs. Criterio externo*. Cada subescala contiene 6 preguntas gradualmente ordenadas a lo largo de la autodeterminación, de tal forma que las 3 primeras preguntas comienzan con el polo intrínseco y las 3 últimas preguntas con el polo extrínseco (de motivación intrínseca a desmotivación).

SRQ-A es otro instrumento (Ryan & Connell, 1989) desarrollado dentro del paradigma de la teoría de la autodeterminación. Este cuestionario mide la motivación preguntando a los alumnos sobre los motivos de su participación en el entorno escolar. Hay cuatro grupos de preguntas en SRQ-A que son: 1) *¿Por qué hago mi tarea?*, 2) *¿Por qué hacer mi trabajo en clase?*, 3) *¿Por qué trato de responder a preguntas difíciles en clase?* Y 4) *¿Por qué trato de sacar buenas notas en la escuela?* Cada pregunta es seguida por ocho ítems (afirmaciones o expresiones que indican posibles respuestas a la pregunta planteada) similares que enumeran las posibles razones de su compromiso en relación con las dimensiones de tareas y el refuerzo de los entornos de aprendizaje. Los 32 ítems en SRQ-A constan de cuatro subescalas de motivación (regulación externa, introyección, la identificación y la motivación intrínseca).

Por otro lado, el instrumento CIPQ (Chow & Law, 2005) fue diseñado para los alumnos que participan en clases de aprendizaje colaborativo apoyado por ordenador (CSCL). Sus 24 ítems (afirmaciones o expresiones que indican posibles respuestas a la pregunta donde el alumno indica el grado con el que está de acuerdo) fueron modificados a una versión de 20 ítems al verificar la validez del instrumento. Las dos primeras preguntas con sus correspondientes respuestas fueron adoptadas desde el Cuestionario de Autorregulación Académica (SRQ-A). A cada ítem se le asocia un factor dependiendo de su significado. Cinco factores fueron etiquetados como factor del proyecto de trabajo, factor de aprendizaje social, factor de tareas, factor de refuerzo y factor de la presión social. Para obtener respuestas específicas a los contextos de aprendizaje colaborativo ha sido desarrollada una tercera pregunta (*¿por qué participas en el trabajo del proyecto?*) con sus 12 ítems de respuesta, donde cuatro elementos son de aspecto de tareas y el resto son de aspecto social de las situaciones de aprendizaje.

Por otra parte, la escala específica de frustración (*Online Collaborative Learning Experiences Frustration Questionnaire – OCLEFQ*) descrita en (Capdeferro & Romero, 2012) mide el grado de frustración que experimentan los alumnos en las tareas de colaboración. La escala se divide en seis secciones. La sección 1 está compuesta por preguntas relacionadas con el aprendizaje y la situación de los alumnos. En las secciones 2 y 3 se recoge información sobre las experiencias previas de los alumnos en el aprendizaje colaborativo en línea y sus actitudes y concepciones sobre el trabajo en equipo. En la sección 4, se pide a los alumnos que informen sobre sus propias experiencias frustrantes a través de una pregunta abierta. La sección 5 fue diseñada para centrarse en las fuentes de frustración con el fin de determinar si había diferencias entre los resultados reportados por las propias experiencias de los alumnos y los que fueron observados por los demás. La sección 6 constaba de seis preguntas que evalúan la frustración del alumno y cómo afecta a su percepción en la participación en la formación apropiada que satisfaga sus ambiciones personales y profesionales.

2.6.3 Trabajos Relacionados con la Evaluación de Motivación

Revisando los trabajos para analizar las implicaciones y relaciones de la motivación con el uso de herramientas CSCL, nos encontramos con los trabajos de Rienties y sus colaboradores (Rienties, et al., 2008), los cuales se llevaron a cabo en un curso de verano, vía Internet, para una asignatura de economía. Este trabajo usó como instrumentos: el Análisis de Contenido, el Análisis de Redes Sociales y la Escala de Motivación Académica (AMS) con el fin de examinar el impacto de los tipos de motivación en el aprendizaje virtual. Otro trabajo en el área de economía (Rienties, et al., 2012) se centró en estudiar la relación que tienen las técnicas de *scaffolding* y la motivación, es decir, la relación de la motivación con el grado de libertad que tiene el alumno para realizar las tareas, y éste está regulado a su vez por el nivel de *scaffolding* aplicado. La motivación es analizada en varias dimensiones (tanto la motivación intrínseca como extrínseca) y como instrumento usan la Escala de Motivación Académica (AMS) utilizando un grupo de control (totalmente autónomos) y grupos experimentales (dirigidos por *scaffolding*). Además realizaron un Análisis de Contenido y Análisis de Redes Sociales.

Otra experiencia descrita en (Rienties, et al., 2009) en el mismo ámbito (economía) y en un entorno de aprendizaje en red, analizó la influencia de la motivación en las contribuciones de los alumnos y en el diálogo. Para este análisis se utilizó la escala AMS para medir la motivación, el

Análisis de Contenido y SNA para el Análisis de Redes Sociales. Los resultados de este análisis indicaron que los alumnos altamente motivados intrínsecamente se convierten en contribuyentes centrales y prominentes en el discurso cognitivo, contribuyendo en la tarea más que otros tipos de alumnos. Mientras que los alumnos altamente motivados extrínsecamente contribuyen significativamente menos en las contribuciones sociales.

El trabajo descrito en (Tapola, et al., 2001) tenía como objetivo analizar la relación entre la participación de los alumnos en la discusión usando la herramienta colaborativa VWS para el intercambio de información en una asignatura de historia y su orientación motivacional. Como instrumento, se utilizó el Análisis de Redes Sociales y un cuestionario para evaluar la orientación motivacional de los estudiantes. El cuestionario contenía cinco ítems (afirmaciones o expresiones que indican posibles respuestas a la pregunta donde el alumno indica el grado con el que está de acuerdo) para evaluar el enfoque de los alumnos en aprender cosas nuevas y adquirir conocimientos. Los resultados indicaron que el nivel de orientación de aprendizaje de los alumnos está estrechamente asociado con la intensidad de su participación en CSCL. Es decir, los alumnos con alta orientación al aprendizaje (más motivados) participan en CSCL con más intensidad.

Howley y otros (2009) usaron la Escala de Motivación Académica (AMS), unos *test* de conocimiento y un cuestionario colaborativo para estudiar si la motivación tiene implicación en la percepción del entorno y del grupo. Para ello, se diseñó un experimento exploratorio para determinar la influencia de la motivación de los alumnos sobre sus interacciones en un entorno de aprendizaje colaborativo utilizando el entorno de colaboración de *Virtual Math Teams* (VMT). Los alumnos valoran la percepción de ellos mismos, de su colaboración o de sus compañeros. Las conclusiones que arrojaron fueron que la motivación del alumno puede afectar a la percepción de lo que hacen los compañeros en el proceso de colaboración y a la percepción del entorno colaborativo.

Eales y sus colegas (2002) estudiaron la influencia de los elementos del entorno sobre la motivación. Se comparó la interacción resultante entre diferentes escenarios de aprendizaje como son la oficina, el colegio y la universidad. Como instrumento de medida se utilizó la opinión de los alumnos sobre la colaboración. Además, distinguieron la auténtica motivación centrada en la evaluación y la táctica de la escolarización, destacando la propiedad del problema de aprendizaje del alumno como una forma particularmente poderosa de motivación.

Otro trabajo ha desarrollado una experiencia con alumnos de un Master universitario en la tarea de escritura colaborativa con el objetivo de medir el grado de frustración de los alumnos (Capdeferro & Romero, 2012) . En este trabajo, se enumeraron los aspectos que hacen que los alumnos experimenten frustración como la colaboración asimétrica entre los miembros del grupo, las barreras que ponen los sistemas Online colaborativos en la comunicación entre los alumnos o el desequilibrio en el nivel de compromiso.

Por otro lado, el trabajo descrito en (Marcos, et al., 2009), para la creación de contenidos Web, describe una herramienta orientada a roles cuyo nombre es *Role-AdaptIA* que aplica el Análisis de Redes Sociales (SNA) como una técnica específica del análisis de interacción y como un instrumento para detectar y ayudar a resolver situaciones problemáticas en escenarios de aprendizaje colaborativo apoyado por ordenador (CSCL). Esta herramienta envía una advertencia al profesor sobre el nuevo papel detectado, como por ejemplo un alumno aislado o un grupo de trabajo que no está colaborando adecuadamente. En estas situaciones las intervenciones posteriores del profesor mejoran las actividades de colaboración.

En el ámbito de la motivación sobre el aprendizaje de la informática, solamente se ha podido encontrar un trabajo, el cual se centra en el sistema colaborativo MoCAS (Serrano-Cámara, et al., 2013) que está diseñado para el aprendizaje de conceptos básicos de programación como pueden ser ámbito y visibilidad de variables. Como instrumento, se utilizaron la escala AMS y los test (pre-post) de eficiencia educativa. En este trabajo se presenta una evaluación de la motivación de los alumnos en cuatro enfoques pedagógico: esquema tradicional, aprendizaje colaborativo, aprendizaje colaborativo guiado por CIF (un marco de diseño de instrucción para el aprendizaje colaborativo) y el aprendizaje colaborativo guiados por CIF y apoyado por la herramienta MoCAS. Los resultados mostraron que los alumnos estaban más motivados por el uso del enfoque de instrucción CIF y la herramienta colaborativa MoCAS.

2.7 Estilos de Aprendizaje

En esta sección se recogen algunos de los diversos modelos de estilos de aprendizaje, centrándose en el modelo de Felder-Silverman.

2.7.1 Modelos de Estilos de Aprendizaje

Todas las personas perciben y adquieren información de manera distinta. Según Alonso y sus colegas (1994) no sólo percibimos de forma diferente, también interaccionamos y respondemos a los ambientes de aprendizaje de manera distinta. El concepto de estilos de aprendizaje se refiere a nuestras preferencias para percibir y adquirir la información, de cómo nuestra mente procesa la información y cómo es influida por nuestras percepciones. Keefe (1979) define los estilos de aprendizaje como los rasgos cognitivos, afectivos y fisiológicos que sirven como indicadores relativamente estables, de cómo los alumnos perciben interacciones y responden a sus ambientes de aprendizaje. Kolb (1984) lo describe como "algunas capacidades de aprender, que se destacan por encima de otras, como resultado del aparato hereditario de las experiencias vitales propias y de las exigencias del medio ambiente actual".

Existen diversos modelos y teorías sobre estilos de aprendizaje, de los cuales destacamos los siguientes modelos:

- **Modelo de Kolb.** Este modelo trata de explicar cómo asimilamos la información, solucionamos problemas y tomamos decisiones. Kolb plantea el aprendizaje como una secuencia cíclica de cuatro momentos que expone sintéticamente en la siguiente figura: existen dos dimensiones principales en el proceso de aprendizaje; la dimensión Abstracto-Concreto que es cómo percibimos la nueva información y la dimensión Activo-Reflexivo que explica cómo procesamos lo que percibimos. Según Kolb (1981), las personas se sitúan en cuatro estilos de aprendizaje: Divergente, Asimilador, Convergente y Acomodador. Los divergentes son los mejores en la visualización de situaciones concretas desde muchos puntos de vista diferentes. Asimiladores pueden procesar de manera lógica y organizar una amplia gama de información. Los convergentes son los mejores en la búsqueda de usos prácticos para las ideas y teorías. Acomodadores a menudo se basan en gran medida en la información de los demás y toman acciones basadas en el instinto más que el análisis lógico. Honey y Mumford (1986) vinculan las motivaciones y expectativas que influyen en el aprendizaje partiendo del análisis de la teoría de Kolb y creando cuatro estilos: Activo (actuar), Reflexivo (reflexionar), Teórico (teorizar) y Pragmático (experimentar).

- **Modelo de Felder & Silverman.** Clasifica los estilos de aprendizaje a partir de cinco dimensiones: Procesamiento (Activo/Reflexivo), Percepción (Sensorial/Intuitivo), Input (Visual/Verbal), Entendimiento (Secuencial/Global) y Organización (Inductivo/Deductivo). Este Modelo se centra en los aspectos de los estilos de aprendizaje significativas en la educación en ingeniería.
- **Modelo de la Programación Neurolingüística.** También llamado Visual-Auditivo-Kinestésico (VAK). Los alumnos visuales aprenden mejor de las imágenes o texto escrito, los alumnos auditivos prefieren la palabra hablada y los alumnos kinestésicos piensan en términos de acción y el movimiento corporal (BECTA, 2005).
- **Modelo de Inteligencias Múltiples de Gardner.** Gardner define a la inteligencia como un conjunto de capacidades que permiten que una persona resuelva problemas y/o forme productos que son de importancia en su vida. Gardner (1993) propone 7 inteligencias. Gardner plantea la hipótesis de que estas siete inteligencias generalmente operan juntos y raramente funcionan de forma independiente. Las inteligencias se utilizan de forma simultánea, por lo general se complementan entre sí a medida que desarrollamos habilidades o resolvemos problemas:
 1. Lógico-Matemática. La capacidad para detectar patrones, razonamiento deductivo y lógico.
 2. Lingüística. La habilidad de usar palabras de manera efectiva.
 3. Espacial. La capacidad de manipular y crear imágenes mentales con el fin de resolver los problemas.
 4. Musical. La capacidad de leer, comprender y componer tonos musicales, tonos y ritmos.
 5. Corporal-Kinestésica. La capacidad de utilizar la mente para controlar uno de los movimientos corporales.
 6. Interpersonal. La capacidad de comprender los sentimientos y las intenciones de los demás.
 7. Intrapersonal. La capacidad de comprender los propios sentimientos y motivaciones.
- **Modelo de hemisferios cerebrales.** El cerebro humano se divide en dos hemisferios. Según Sperry (1961), ambos lados del cerebro tienden a dividirse las principales funciones intelectuales. El hemisferio izquierdo procesa la información de manera secuencial, lineal y lógica mientras que el hemisferio derecho es el sintetizador, el de la estructura total y la

imaginación, es intuitivo en vez de lógico. Normalmente para aprender bien, debemos usar ambos hemisferios pero tendemos de usar uno más que el otro, este hecho determina las habilidades cognitivas de cada individuo.

- **Modelo de Cuadrantes Cerebrales.** El modelo de cuatro lados de estilos de pensamiento de Ned Hermann (1988) metafóricamente atribuye a cuatro regiones del cerebro. La creatividad es un proceso que involucra los cuatro cuadrantes. Estos cuatro cuadrantes se pueden caracterizar como:
 1. Cortical Izquierdo. Orientado al presente.
 2. Límbico Izquierdo. Orientado a los procesos.
 3. Límbico Derecho. Orientado a las personas.
 4. Cortical Derecho. Orientada al futuro.

2.7.2 Modelo de Felder-Silverman

Este enfoque se ha defendido como un ambiente de aprendizaje eficaz para la enseñanza de la ingeniería. La comprensión de los resultados de la prueba de estilos de aprendizaje de Felder-Silverman puede ayudar al profesor a encontrar formas adecuadas para presentar el material a los alumnos. Sin embargo, los resultados no se deben utilizarse como base para recomendar planes de estudios para los alumnos, ya que hacerlo puede limitar sus experiencias de desarrollo que les impiden llevar a cabo esfuerzos en los que pueden encontrar el éxito (Felder & Spurlin, 2005). Sin embargo, según Alonso y sus colegas (1994), cuando el tipo de enseñanza es acorde al estilo de aprendizaje del alumno, éste aprende con mayor efectividad. Asimismo, Cassidy (2004) afirma que la manera en que los individuos eligen o se inclinan a acercarse a una situación aprendizaje tiene un impacto en el rendimiento y el logro de los resultados del aprendizaje.

El modelo de Felder-Silverman consta de 8 estilos de aprendizaje.

1. Activo. Mejora la retención y la comprensión de la información mediante el debate y o explicación a otros. Le favorece las actividades de grupo en la que los miembros expliquen los temas entre sí y encontrar maneras de aplicar o utilizar la información.
2. Reflexivo. Prefiere pensar en el material primero. Le favorece revisar periódicamente lo que se ha leído y la redacción de un resumen de las lecturas o apuntes de clase.

3. Sensorial. Disfruta de cursos que tienen conexiones con el mundo real. Le favorece la conexión de información a las aplicaciones del mundo real.
4. Intuitivo. No le gusta los cursos que requieren memorización y cálculos rutinarios. Le favorece la búsqueda de interpretaciones o teorías que vinculan los hechos. Y tener cuidado al leer la pregunta completa antes de responder y volver a comprobar el trabajo para evitar errores por descuido.
5. Visual. Recuerda lo que ve; como imágenes, diagramas, diagramas de flujo, las manifestaciones. Le favorece encontrar o dibujar diagramas, bocetos, esquemas, fotografías, videos, materiales para describir curso. Y el uso de mapas conceptuales para organizar visualmente los puntos clave.
6. Verbal. Comprende mejor de las explicaciones escritas y habladas. Le favorece los resúmenes de escritura y trabajar en grupos para escuchar explicaciones de los compañeros.
7. Secuencial. Ganancia de la comprensión en pasos lineales y lógicos. Le favorece esbozar material de clase en un orden lógico y relacionar los nuevos temas a las cosas ya conocidas para fortalecer las habilidades de pensamiento.
8. Global. Aprende en grandes saltos, absorbiendo al azar material hasta que de repente "hacer las cosas". Le favorece repasar a través todo el capítulo para obtener una visión general antes de comenzar a estudiar la información específica y relacionar el tema con cosas ya conocidas

Según De Bello(2006), para elegir el modelo correcto o instrumento para la investigación, éste debe ser evaluado con respecto a su fiabilidad y validez. Por eso hemos usado el modelo de Filder-Silverman en nuestra investigación por ser validado (Zywno, 2003), además de popular entre los educadores de ingeniería, influyente y probado.

Capítulo 3: Sistemas GreedEx y GreedExCol

Este capítulo se centra en los sistemas GreedEx y GreedExCol. En primer lugar se presentan los sistemas antecedentes AMO; SEDA, para la experimentación interactiva con algoritmos voraces; y trabajos relacionados. En segundo lugar, se describe el sistema TuMiST. En tercer lugar, se describe la herramienta GreedEx y por último, presentamos su ampliación colaborativa GreedExCol.

3.1 Antecedentes y Trabajos Relacionados

El trabajo realizado tenía como antecedentes un método didáctico para el aprendizaje activo de los algoritmos voraces que fue concebido como formado por 2 componentes, un método experimental (Velázquez-Iturbide & Pérez-Carrasco, 2009a) y los asistentes interactivos AMO y SEDA que daban soporte al método para los problemas de la mochila (Cormen, et al., 2001)(Sahni, 1995) y selección de actividades (Cormen, et al., 2001). En este apartado, se repasan los sistemas y trabajos antecedentes que han sido la base y punto de partida para el desarrollo de mi trabajo de tesis.

Los objetivos de aprendizaje de los algoritmos voraces pueden expresarse en términos de la taxonomía de Bloom (Anderson, et al., 2001). Este marco clasifica los objetivos de aprendizaje y establece una jerarquía de seis niveles para aumentar el grado de experiencia cognitiva del alumno:

1. Nivel de conocimientos. El alumno es capaz de reconocer o recordar la información sin necesidad de tener ningún tipo de conocimiento de, o la capacidad para el razonamiento sobre, su contenido.
2. Nivel de comprensión. El alumno es capaz de comprender y explicar el significado de la información recibida.
3. Nivel de aplicación. El alumno es capaz de seleccionar y utilizar los datos y los métodos para resolver una tarea determinada o problema.
4. Nivel de análisis. El alumno es capaz de distinguir, clasificar y relacionar una hipótesis y la evidencia de la información proporcionada, así como la descomposición de un problema en sus partes.

5. Nivel de síntesis. El alumno es capaz de generalizar las ideas y aplicarlas para resolver un nuevo problema.
6. Nivel de evaluación. El alumno es capaz de comparar, criticar y evaluar los métodos o soluciones para resolver un problema, o para elegir el mejor método o solución.

En el caso de los algoritmos voraces, puede proponerse que el alumno deba ser capaz de (Velázquez-Iturbide & Pérez-Carrasco, 2009a):

1. Conocer los fundamentos de la técnica (nivel de comprensión).
2. Comprender algoritmos voraces clásicos, por ejemplo, el algoritmo de Dijkstra (nivel de comprensión).
3. Descubrir una función de selección óptima para un problema de optimización (nivel de aplicación o síntesis).
4. Programar un algoritmo voraz (nivel de aplicación o síntesis).
5. Demostrar la optimalidad de una función de selección voraz (nivel de aplicación o síntesis).

Estos objetivos de aprendizaje no son fáciles de lograr con la organización de los materiales de los algoritmos voraces que se pueden encontrar en la mayoría de los libros de texto. En los capítulos de algoritmos voraces, se suele identificar los principales elementos, ilustrar la técnica con una plantilla de alto nivel y resolver una serie de problemas. Para cada problema, una función de selección óptima voraz se identifica y su optimalidad se demuestra. Por último, el algoritmo se codifica de forma iterativa y se analiza su complejidad.

En resumen, a pesar de la aparente sencillez de algoritmos voraces, sus objetivos de aprendizaje son muy exigentes. Como consecuencia de ello, el profesor habitualmente da clases de una manera pasiva, apenas dejando lugar para el aprendizaje activo (Velázquez-Iturbide, 2013) (Velázquez-Iturbide, 2011).

3.1.1 Método Experimental

El método didáctico tiene cuatro componentes en su formulación final (Velázquez-Iturbide, 2013): un método experimental orientado a descubrir funciones de selección óptimas, los sistemas interactivos de apoyo (GreedEx y GreedExCol), la organización de clases y prácticas (específicamente de los algoritmos voraces), y materiales de apoyo (apuntes). En este apartado nos centramos en el método didáctico. Los sistemas se presentan en las secciones 3.1.2 (los sistemas antecedentes), 3.3 y 3.4 (sistemas GreedEx y GreedExCol). El método didáctico completo se presenta en la sección 3.1.5.

El método didáctico planteado (Velázquez-Iturbide, 2013) (Velázquez-Iturbide & Debdí, 2011) se basa en el aprendizaje por descubrimiento y el aprendizaje basado en problemas. Se da un problema de optimización al alumno y un número de funciones de selección que pueden caracterizar diferentes algoritmos voraces y entonces se requiere encontrar el subconjunto de funciones de selección que conducen a algoritmos voraces óptimos. En general, el conjunto de posibles funciones de selección no es muy grande, ya que se limitan a los valores de algunos parámetros o valores significativos derivados de ellas, clasificadas en orden decreciente o creciente. Esta búsqueda de optimalidad no se logra mediante pruebas formales sino por medio de la experimentación que es mucho más fácil y más atractivo para el alumno (Velázquez-Iturbide, et al., 2012). Veámoslo con el problema de selección de actividades (Cormen, et al., 2001):

Supongamos que tenemos un conjunto $S = \{a_1, a_2, \dots, a_n\}$ de n actividades que deseen hacer uso de un recurso, como una sala de conferencias por ejemplo, que debe ser utilizado por una sola actividad a la vez. Cada actividad a_i tiene un tiempo de inicio c_i y un tiempo de finalización f_i , donde $0 \leq c_i < f_i < \infty$. Si se selecciona la actividad a_i , ocuparía el intervalo semiabierto $[c_i, f_i)$. Las actividades a_i y a_j son compatibles si los intervalos $[c_i, f_i)$ y $[c_j, f_j)$ no se superpongan (es decir, a_i y a_j son compatibles si $c_i \geq f_j$ o $c_j \geq f_i$). El problema es seleccionar un subconjunto de tamaño máximo de actividades mutuamente compatibles.

Por ejemplo, considere el siguiente conjunto S de actividades que hemos ordenado por orden de tiempo creciente de finalización:

Tabla 4. Un ejemplo del problema de selección de actividades

I	0	1	2	3	4	5	6	7	8	9	10
c_i	1	3	4	5	3	5	6	8	8	2	12
f_i	4	5	6	7	8	9	10	11	12	13	14

El alumno debe tratar de descubrir si existe una función de selección óptima (voraz) que resuelve el problema. Una serie de funciones de selección de actividades son propuestas. En general, las funciones de selección se basan en los parámetros del problema ordenando las actividades en orden creciente o decreciente. Pueden ser razonables las funciones de selección basados en los siguientes seis criterios:

- Orden Creciente/Decreciente de comienzo de la actividad.
- Orden Creciente/Decreciente de fin de la actividad.
- Orden Creciente/Decreciente de la duración de la actividad.

Si aplicamos las seis funciones de selección a los datos que figuran en la Tabla 4, se obtienen los resultados mostrados en la Tabla 5.

Tabla 5. Resultado de la aplicación de diferentes funciones de selección

FUNCIÓN DE SELECCIÓN	ACTIVIDADES SELECCIONADAS	NUMERO DE ACTIVIDADES
Orden creciente de comienzo	2,6,10	4
Orden decreciente de comienzo	10,8,3,1	4
Orden creciente de fin	0,3,7,10	4
Orden decreciente de fin	10,8,4	3
Orden creciente de duración	1,3,10,7	4
Orden decreciente de duración	9	1

Estos resultados permiten descartar la cuarta y sexta función de selección. Ahora debemos experimentar con las cuatro restantes funciones de selección con nuevos datos de entrada. La repetición de este experimento debe ser tantas veces como sea necesario, vamos a llegar a un estado en el que sólo se quedan las funciones de selección factibles. Siguiendo nuestro ejemplo, hemos utilizado sucesivamente cuatro ejecuciones con nuevos datos de entrada y se obtuvieron los resultados que se resumen en la Tabla 6. Sólo incluimos el número de actividades seleccionadas en cada caso.

Tabla 6. Resultados sobre diferentes datos de entrada

FUNCIÓN DE SELECCIÓN	EJECUCIÓN	EJECUCIÓN	EJECUCIÓN	EJECUCIÓN
	1	2	3	4
Orden creciente de comienzo	4	3	2	-
Orden decreciente de comienzo	4	3	3	3
Orden creciente de fin	4	3	3	3
Orden decreciente de fin	3	-	-	-
Orden creciente de duración	4	3	3	3
Orden decreciente de duración	1	-	-	-

Observe que las tres funciones de selección permanecen después de las cuatro ejecuciones. Podríamos seguir experimentando con nuevos datos de entrada o parar y tratar de demostrar formalmente la optimización de estas tres funciones de selección. Realizar varias pruebas puede revelar errores pero no asegura la corrección. La corrección sólo puede ser probada por medio de la verificación. En nuestro caso, la experimentación puede revelar las funciones de selección no óptimas, pero por sí solo no garantiza haber identificado las funciones de selección óptimas. Cada función de selección restante debe ser individualmente objeto de una prueba de optimalidad. Por ejemplo, la Tabla 6 muestra tres funciones de selección óptimas, sin embargo, uno de ellos (orden creciente de duración) no es óptima.

3.1.2 Sistemas de Apoyo

En esta sección se describen los sistemas de apoyo para el aprendizaje activo de algoritmos voraces AMO y SEDA (Velázquez-Iturbide, et al., 2009c). Primero, describimos sus características generales, después se detalla cada uno por separado.

3.1.2.1 Características Generales

La interfaz gráfica de los dos sistemas de apoyo antecedentes presenta un diseño similar que consta de los siguientes elementos: una barra de menús, una barra de herramientas o iconos y tres paneles que cumplan las siguientes funcionalidades:

- Panel de visualización que se encarga de la representación gráfica.
- Panel de teoría algoritmo que contiene información relativa al algoritmo, en concreto el enunciado del problema y pseudocódigo del algoritmo.

- Panel de resultados que muestra los resultados de las ejecuciones.

El alumno experimenta con el ayudante interactivo siguiendo el método experimental descrito en el sub-apartado anterior:

1. Lee y entiende el enunciado del problema y el algoritmo voraz genérico escrito en pseudocódigo.
2. Genera un conjunto de datos de entrada, ejecuta el algoritmo voraz utilizando todas las funciones de selección y comprueba cuál de estos resultados óptimos y cuáles no. Las funciones de selección que no producen resultados óptimos se descartan, de los cuales se puede decir que el alumno ha encontrado un contraejemplo de su optimalidad.
3. Repite el paso 2 como sea necesario hasta obtener pruebas suficientes de la optimalidad de las funciones de selección restantes.

Algunas características destacadas de los sistemas AMO y SEDA son las siguientes:

- Introducción de datos de entrada de varias maneras (generación automática, cargar desde un fichero o por teclado)
- Funciones de selección para solventar el problema de diferentes maneras.
- Ejecución del algoritmo paso a paso o de golpe.
- Visualización de los datos introducidos y de los resultados mediante gráficos intuitivos y tablas.
- Visualización de resultados mediante varias tablas.
- Exportación de datos de entrada como los resultados utilizados para su posterior uso.
- Internacionalización.

3.2.1.2 Sistema AMO

El asistente interactivo AMO es una herramienta para el aprendizaje del algoritmo de la mochila (Cormen, et al., 2001)(Sahni, 1995). El planteamiento del problema de la mochila es el siguiente:

Dada una mochila con una capacidad C , que es el peso máximo que puede soportar y n objetos, cada objeto i , $0 < i < n$ tiene un peso p_i y un beneficio b_i . El objetivo es llenar la mochila de forma que se maximice el beneficio total. Los objetos se pueden partirse en trozos pequeños, de forma que podemos tomar el objeto entero o una fracción de él; si se mete una fracción $x_s[i]$, tal que $0 \leq x_s[i] \leq 1$, del objeto i de la mochila, entonces se consigue un beneficio $x_s[i] * b_s[i]$. Es decir, maximizar $\sum x_s[i] * b_s[i]$. Una solución del problema sería un vector de objetos x_1, x_2, \dots, x_n que nos indica las fracciones de los objetos que deben estar en la mochila mientras la función objetivo sería el beneficio total de los fracciones de los objetos que están en la mochila.

Por ejemplo, sea una mochila con una capacidad igual a $C = 264$ y 5 objetos con $p_s = \{79, 52, 78, 18, 50\}$ y $b_s = \{57, 38, 38, 18, 41\}$. Tras ejecutar el algoritmo usando la función de selección de orden creciente de beneficio obtendremos el beneficio de 182.62, mientras si ejecutemos el algoritmo con la función de selección de orden decreciente beneficio/peso obtendremos el beneficio máximo de 185.66.

La Ilustración 2 muestra el aspecto de la interfaz de usuario de AMO que dispone de tres zonas, una para el panel del problema que muestra los objetos en la parte izquierda de manera que el ancho de los objetos representa el peso y la altura representa el beneficio. En la parte derecha se muestra la mochila donde su anchura representa su capacidad. En la parte inferior se encuentra el panel de teoría en la izquierda y el panel de resultados en la derecha. Además de la barra de menús, una barra de herramientas con iconos de acceso a las opciones más frecuentes.

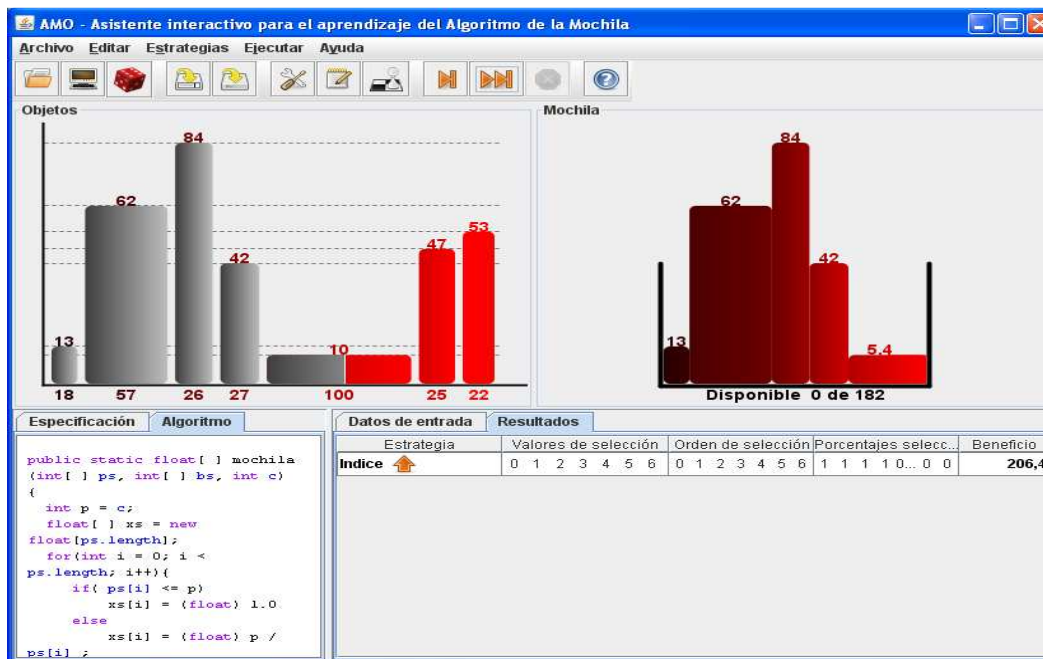


Ilustración 2. Interfaz de usuario de AMO

Al ejecutar AMO, el alumno podrá consultar el panel de teoría y del algoritmo, como podrá generar los datos de entrada y seleccionar una de las funciones de selección disponibles para la ejecución del algoritmo. Las funciones de selección integradas en el sistema AMO son:

- Orden creciente/decreciente de índice.
- Orden creciente/decreciente de peso.
- Orden creciente/decreciente de beneficio.
- Orden creciente/decreciente de beneficio/peso.
- Orden creciente/decreciente de peso/beneficio.

Obsérvese que las dos primeras funciones de selección equivalen a una selección al azar. Sin embargo, se han añadido para aumentar la oferta de funciones de selección disponibles para los alumnos.

Al ejecutar el algoritmo, los objetos seleccionados se marcarán en gris en el panel de datos de entrada y se introducirán con su color original dentro de la mochila. Para resaltar el orden de los objetos según la función de selección elegida, los objetos se dibujaran en un mismo color, su orden

vendrá indicado por su tonalidad de más a menos oscura, de forma que los objetos que ocupen las primeras posiciones tendrán tonos más oscuros, mientras que los últimos objetos tendrán tonos más claros.

En el panel de resultados, la primera tabla contiene un resumen de los datos de entrada y en la segunda, los resultados de la ejecución del algoritmo de la mochila, en el que se indicará: la función de selección ejecutada, los valores de selección, el orden de selección de los objetos, el porcentaje seleccionado de cada objeto y por último el beneficio obtenido.

3.1.2.2 Sistema SEDA

El asistente interactivo SEDA es una herramienta para el problema de selección de actividades (Cormen, et al., 2001). El planteamiento de este problema se encuentra en la sección 3.1.1.

La interfaz global de SEDA es igual al de AMO, la diferencia es la visualización del panel de datos, las funciones de selección y los datos de entrada, dado que SEDA es un asistente interactivo para el aprendizaje del problema de selección de actividades. La Ilustración 3 muestra la interfaz de SEDA con una ejecución de un ejemplo.

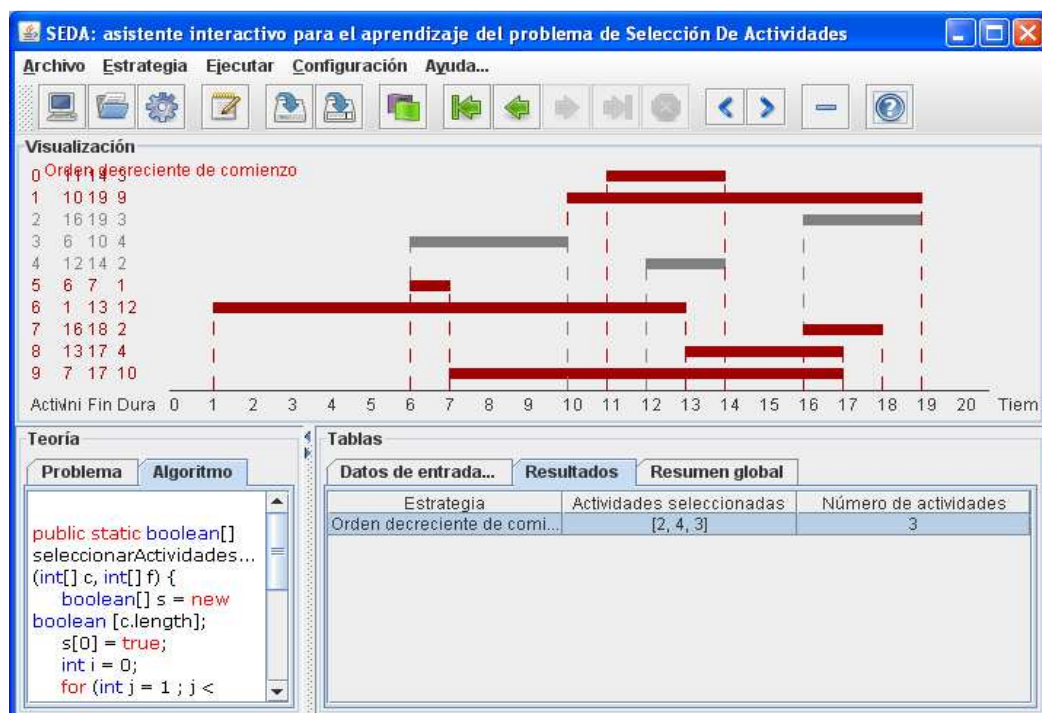


Ilustración 3. Interfaz de usuario de SEDA

Cada actividad se representa mediante una barra comprendida entre los instantes de comienzo y fin sobre un eje temporal horizontal. Cada actividad seleccionada se colorea de gris si es válida o de rojo si se solapa con alguna seleccionada antes. Al seleccionar una función de selección se colorean las actividades con tonos según la función de selección seleccionada, es decir las actividades con un color más oscuro serán elegidas antes de las actividades coloreadas con un tono claro.

SEDA ofrece ocho funciones de selección:

- Índice de la actividad en orden creciente/decreciente.
- Comienzo de la actividad en orden creciente/decreciente.
- Fin de la actividad en orden creciente/decreciente.
- Duración de la actividad en orden creciente/decreciente.

En el panel de resultados, la primera tabla contiene un resumen de los datos de entrada y en la segunda los resultados, donde se muestra el beneficio producido por las distintas funciones de

selección ejecutadas. Cada fila contiene la función de selección, las actividades seleccionadas y el valor de la función objetivo (número de actividades). Una nueva tabla contiene un resumen global de todas las ejecuciones realizadas.

3.1.2.3 Sistema GreedEx

AMO y SEDA son sistemas para los problemas de la mochila y selección de actividades respectivamente. Sin embargo, el principal problema de estos sistemas era que cada sistema únicamente soportaba un problema resoluble vorazmente. Como consecuencia, su interés educativo para los profesores era limitado y el mantenimiento de varios sistemas era problemático. GreedEx (Velázquez-Iturbide, et al., 2010) es un sistema que ha integrado los sistemas AMO y SEDA. La Ilustración 4 muestra una interfaz de usuario de la herramienta GreedEx.

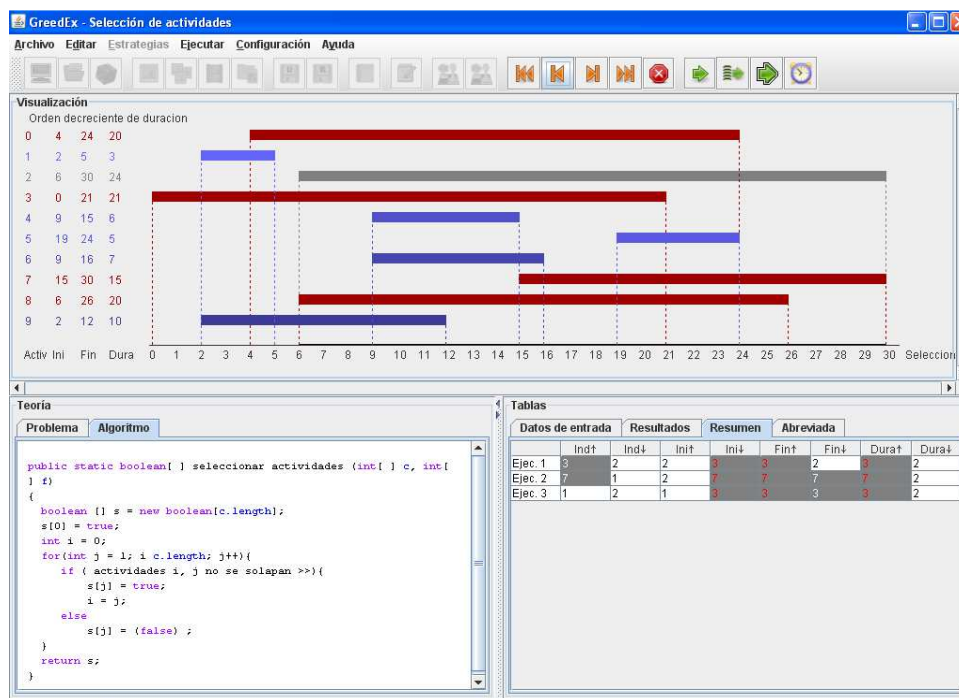


Ilustración 4. Interfaz de usuario de GreedEx

GreedEx utiliza la misma visualización básica para todas las funciones de selección definidos para un algoritmo dado. La visualización se adapta a cada función de selección en particular mediante el uso de tonos de color para ordenar implícitamente los candidatos según la función de selección elegida (una funcionalidad ya desarrollada en los sistemas AMO y SEDA).

El usuario puede elegir el problema a estudiar en un menú fijo. Los elementos que varían con cada problema son:

- Diálogos de entrada o modificación de datos.
- Panel del problema (pestañas de explicación y de algoritmo).
- Panel de visualización.
- Menús de las funciones de selección a ejecutar (y su implementación).
- Formato de las tablas de datos de entrada y de resultados, así como leyendas de las funciones de selección en las tablas de resultados, de resumen y abreviada.
- Configuración de la visualización.

Se ha añadido una cuarta tabla, llamada tabla abreviada que muestra los resultados en porcentaje obtenidos por cada función de selección. También se ha incluido una nueva función de ejecución de las funciones de selección sobre un número muy alto de datos de entrada (“ejecución intensiva”). El usuario puede elegir entre ejecutar sobre un número dado de datos de entrada o durante un cierto tiempo. Como la ejecución intensiva puede durar bastante, el usuario puede realizar algunas funciones en paralelo sin que el asistente se quede bloqueado.

También hay que comentar que AMO, SEDA y GreedEx se sometieron a tres evaluaciones de usabilidad por usuarios, que permitieron refinar sus funciones y añadir otras nuevas, de forma que soportaran mejor el método experimental. No se describen estas evaluaciones pero son similares en estructura a las realizadas en el seno de esta tesis y presentadas en el Capítulo 4. Puede encontrarse más información en (Velázquez-Iturbide, et al., 2009c) (Velázquez-Iturbide, et al., 2013a).

3.1.3 Análisis de Malentendidos

El objetivo de este análisis fue evaluar el método didáctico y en consecuencia detectar los malentendidos de los alumnos acerca de los algoritmos voraces y eliminarlos. El método didáctico planteado para el aprendizaje activo de algoritmos voraces se ha utilizado durante cinco años en la asignatura “Diseño y Análisis de Algoritmos” de curso de Ingeniería Informática de la universidad Rey Juan Carlos. Como parte de esta experiencia, se analizaron los informes entregados por los alumnos voraces (Velázquez-Iturbide, Aceptado), coincidiendo con tres evaluaciones de usabilidad realizadas en enero de 2009, noviembre de 2009 y noviembre de 2010. Se identificaron varios malentendidos sobre conceptos básicos de optimización que es la base de algoritmos voraces. La

estructura de esta evaluación es similar a las sesiones de evaluación de usabilidad realizadas en este trabajo de tesis y detalladas en el capítulo 4.

Los alumnos tenían que resolver una práctica usando el sistema GreedEx. El profesor presentó algoritmos voraces en dos sesiones anteriores en el aula, utilizando también el sistema GreedEx. Al comienzo de la sesión de la práctica, los alumnos descargaron todos los materiales necesarios: el enunciado, el sistema GreedEx y una plantilla del informe que hay que entregar al final. La práctica contenía el enunciado del problema y una breve descripción de GreedEx. Los alumnos tuvieron que realizar varias tareas ya sea individualmente o en parejas:

1. Utilizar GreedEx para identificar las funciones de selección óptima para el problema de la selección de actividades.
2. Llenar y presentar electrónicamente un breve informe, escrito utilizando una plantilla de Word simple. Los alumnos tenían que identificar las funciones de selección óptimas, dar una justificación informal y proporcionar pruebas, de forma resumida y detallada.
3. Llenar y entregar el cuestionario de opinión (voluntario).

Se han resumido muchos aspectos de cada informe entregado: las funciones de selección propuestas como óptimas, la calidad de su justificación (informal), la calidad de las pruebas presentadas y la calidad del razonamiento global. Además, se registró: el número de ejecuciones llevadas a cabo, el número de ejemplos incluidos, el tamaño de ejemplos y el número de funciones de selección ejecutadas por datos de entrada.

Una consecuencia inesperada de la utilización del método experimental fue el descubrimiento de malentendidos sobre algoritmos de optimización. Se identificaron cuatro conceptos erróneos:

1. Propuesta de funciones de selección de óptimas. En particular, la función de selección de orden creciente de duración (el problema de selección de actividades) debido a la dificultad de descubrir la falta de su optimalidad.
2. La incoherencia en el razonamiento deductivo. Algunos alumnos proponen funciones de selección no óptimas (incluso después de encontrar un contraejemplo) o descartan funciones de selección óptimas (incluso sin encontrar un contraejemplo).

3. Criterios adicionales de optimización. Algunos alumnos proponen un criterio adicional, por ejemplo, maximizar el tiempo de uso del recurso o minimizar el tiempo de espera de las actividades. La razón era que los alumnos consideran "óptimo" como sinónimo de "único".
4. Propuesta de diferentes funciones de selección óptimas para diferentes datos de entrada. En este caso, el error es aún más fuerte, ya que estos alumnos no consideran "óptimo" como sinónimo de "siempre", sino de "particular".

Esto dio la oportunidad de mejorar aún más la enseñanza de algoritmos de optimización para reducir al mínimo el número de ideas erróneas. En consecuencia, se introdujeron varios cambios en el método didáctico. Las intervenciones más evidentes fueron en clases donde se han eliminado conceptos erróneos graves, gracias a una combinación adecuada del método didáctico, software educativo, programación de las clases y sesiones de laboratorio y los materiales de enseñanza.

3.1.4 Refinamiento del Método Didáctico

El método didáctico planteado (Velázquez-Iturbide, 2012) (Velázquez-Iturbide, 2013) es un enfoque innovador compuesto inicialmente por un método experimental y una herramienta para apoyarlo (GreedEx). La herramienta se ha especificado con objetivos pedagógicos explícitos y fue evaluada en cursos sucesivos con respecto a su usabilidad para soportar plenamente el método experimental. Como se ha explicado en el subapartado anterior, también se analizaron los informes de los alumnos con el fin de detectar los conceptos erróneos. El resultado de este trabajo de 5 años fue refinar los elementos del método didáctico.

Para refinar el método didáctico se han añadido una serie de actividades encaminadas a eliminar los malentendidos. En primer lugar, se programó más tiempo al principio del capítulo sobre algoritmos voraces donde se identificaron explícitamente posibles funciones de selección para los problemas de optimización, se formuló el método experimental utilizando GreedEx, se identificaron contraejemplos y se relacionaron (estos experimentos) a la experimentación científica y pruebas de software. En segundo lugar, se elaboraron nuevos apuntes para eliminar los conceptos erróneos, abordando los problemas de optimización, la experimentación y el sistema GreedEx. En tercer lugar, se aumentó el número de sesiones de laboratorio introduciendo una última sesión de laboratorio pidiendo a los alumnos identificar los errores cometidos en sus informes. Además, se introdujo una

sesión de introducción, donde los alumnos experimentaron con un problema más sencillo (el problema de la mochila). De esta manera, los alumnos consiguieron una experiencia práctica con experimentación y se familiarizaron con GreedEx. En cuarto lugar, se introdujo cambios en los informes que tienen que rellenar los alumnos para la práctica. Una plantilla más estructurada con evidencia experimental más clara. Por último, se utilizó GreedEx para mostrar que algunos problemas no pueden ser resueltos de manera óptima con algoritmos voraces, por lo que deben ser resueltos usando otras técnicas de diseño (problema de la mochila 0/1). Para ello, se ha extendido el método experimental a lo largo del curso para introducir los algoritmos de *backtracking* y programación dinámica.

3.1.5 Eficacia Educativa

Una vez refinado el método didáctico, se evaluó con respecto a su eficacia educativa. Esta evaluación (Esteban Sánchez, et al., 2014) se realizó en la asignatura troncal “Diseño y Análisis de Algoritmos” de segundo curso de Ingeniería Informática. El curso tenía quince semanas de duración con dos sesiones por semana, de dos horas de duración. Cinco sesiones se programaron para el tema de los algoritmos voraces. Se consideraron dos grupos: un grupo de control con 18 alumnos matriculados en el campus de Vicálvaro y un grupo experimental de 50 alumnos matriculados en el campus de Móstoles. El método experimental se incluyó principalmente como una introducción a los algoritmos voraces, así las clases de los dos grupos difieren en la primera, segunda y cuarta sesión. La tercera y quinta sesiones fueron sesiones de clases tradicionales dedicadas a algoritmos voraces clásicos (algoritmos de Prim y Kruskal, códigos de Hoffman y el algoritmo de Dijkstra). Los alumnos en el grupo de control siguieron un horario tradicional de las clases y de las sesiones de laboratorio. Sin embargo, en el grupo experimental, en la primera mitad de la segunda sesión, el profesor explicó los principios del método experimental y el uso de GreedEx. En la segunda mitad de la segunda sesión, los alumnos tenían que aplicar el método experimental en el laboratorio para encontrar las funciones de selección óptimas para el problema de la mochila, lo cual es un reto sencillo. Por último, en la cuarta sesión, los alumnos tuvieron que resolver una tarea de descubrimiento que consiste en la búsqueda de las funciones de selección óptimas para el problema de la selección de actividades, que es un problema mucho más exigente.

Los alumnos de ambos grupos completaron una prueba previa al inicio de la primera sesión de la sección del curso de algoritmos voraces. La misma prueba se entregó a los alumnos como un *posttest* al final del quinto período de sesiones. El profesor fue el mismo para ambos grupos. La prueba se componía de cinco preguntas, cada una de las cuales se califican en una escala de 0 (el grado más bajo) al 10 (el más alto). La calificación final fue la media de las calificaciones obtenidas en las cinco preguntas. Las tres primeras preguntas de la prueba abordaron los conceptos básicos de la optimización o de los algoritmos voraces. Las siguientes dos preguntas se referían a la solución óptima de los dos problemas sencillos utilizando la técnica voraz; en la cuarta cuestión, se les preguntó sobre la viabilidad de la solución óptima del problema con algoritmos voraces y en el quinto se les preguntó sobre la optimalidad de una función de selección para un problema particular.

En la evaluación también se pretendió identificar los niveles de aprendizaje a medir. Estos niveles se identificaron claramente en términos de la taxonomía de Bloom. Se midió el aprendizaje en los niveles de conocimiento, comprensión y análisis. Los logros de aprendizaje fueron analizados mediante la notas de los problemas en los niveles de aprendizaje de la taxonomía de Bloom. La evaluación detectó medibles avances en el aprendizaje de los alumnos en el grupo experimental, mientras que estos logros de aprendizaje no se detectaron en el grupo de control. Los resultados confirmaron una mejora en los niveles de conocimiento y de comprensión, pero no en el nivel de análisis. Además, sugieren futuras líneas de investigación como mejorar las habilidades de los alumnos en el análisis de problemas de optimización y algoritmos, probablemente utilizando el método a lo largo de todo un curso de introducción de técnicas de diseño de algoritmos para problemas de optimización.

3.2 Sistema TuMiST

TuMiST (*Tutor Minimum Spanning Tree*) es un sistema interactivo para el aprendizaje de los algoritmos Prim y Kruskal (Debdi, et al., 2010b). Una de sus dificultades técnicas era la visualización de grafos. TuMiST fue el tercer sistema desarrollado en apoyo del método experimental, a partir de la experiencia de AMO y SEDA. En esta sección detallamos el problema de recubrimiento de coste mínimo y describimos la interfaz de la herramienta.

El planteamiento del árbol de recubrimiento de coste mínimo (Cormen, et al., 2001) es el siguiente:

Un árbol de recubrimiento, conexo y con arcos no dirigidos, es un árbol que contiene todos los vértices del grafo, pero contiene solo un subconjunto de los arcos, es decir, solo los necesarios para conectar entre ellos todos los vértices con un solo camino. De hecho, lo que diferencia un grafo de un árbol es que en este último no existen caminos múltiples entre dos nodos, es decir, no hay ciclos. En el caso de que los arcos tengan un peso, se define también el árbol de recubrimiento de coste mínimo, es decir, el árbol de conexión de peso mínimo, donde el peso total es la suma de los pesos de todos los arcos que pertenecen a dicho árbol.

En el algoritmo de Kruskal, la función de selección escoge las aristas por orden creciente de longitud, sin observar la conexión con las aristas seleccionadas antes, teniendo en cuenta solo el hecho de no crear ciclos. El resultado es un bosque de árboles que crece al azar, hasta que finalmente todas las componentes del bosque se fusionan en un único árbol. En el algoritmo de Prim el árbol de recubrimiento mínimo crece de forma única, comenzando por una raíz arbitraria. En cada fase se añade una nueva rama al árbol ya construido; el algoritmo se detiene cuando se alcanzan todos los nodos.

En la Ilustración 5 podemos ver el aspecto del sistema TuMiST. Al igual que los otros sistemas presentados, dispone de una barra de menú, una barra de herramientas con iconos de acceso a las opciones más frecuentes, y tres paneles con información diferente. El panel superior contendrá en la parte izquierda, el grafo inicial que introducirá el usuario y una vez resuelto el problema, su respectivo árbol de recubrimiento en la parte derecha. El panel inferior izquierdo contiene una pequeña explicación del algoritmo del árbol de recubrimiento de coste mínimo. Y en la otra pestaña podemos encontrar dos de los principales pseudocódigos para llegar a una solución óptima. El último panel, en la parte inferior a la derecha, es una tabla que contendrá los resultados una vez que el usuario haya cargado un grafo y haya aplicado algunas de las funciones de selección disponibles, donde se muestra la función de selección ejecutada, el orden de los arcos elegidos y el peso del respectivo árbol de recubrimiento.

The screenshot shows the TuMiST software interface. The main window displays a graph with 5 nodes (1, 2, 3, 4, 5) and 7 edges with weights: (1,2) weight 10, (1,3) weight 5, (2,3) weight 5, (2,4) weight 5, (3,4) weight 1, (3,5) weight 5, and (4,5) weight 5. The interface includes a menu bar (Archivo, Ejecutar, Estrategias, Configurar, Ayuda), a toolbar, and a logo. Below the graph, there is an 'Explicación' section with pseudocode and a 'RESULTADOS' table.

Explicación

Pseudocódigo

A := {Todos los arcos} ; N := {Todos los nodos} ; S := {} // solución
 N' := {n_e} // primer nodo elegido ; C := {} // cola
para cada w ∈ N tal que (n_e, w) ∈ A **hacer**
 insertar(C, (n_e, w))
mientras |N'| < |N|
 e := elige_arco(C, base_estrategia)
 se supone e = (u,v) donde u ∈ N'
 si v ∈ N' **entonces**
 S := S ∪ {e}
 N' := N' ∪ {v}
 para cada w ∈ N tal que (v, w) ∈ A **hacer**
 insertar(C, (v, w))

Monografo Multigrafo

RESULTADOS

Estrategia	Orden de los arcos	Peso
Nodos ↓	(3,5) (3,4) (2,4) (1,3)	27
Arco ↓	(1,3) (1,2) (2,4) (3,5)	36
Nodos ↑	(1,2) (1,3) (2,4) (3,5)	36
Nodos ↓	(3,5) (3,4) (2,4) (1,3)	27
Arco ↑	(3,4) (3,5) (2,4) (1,2)	24
Arco ↓	(1,3) (1,2) (2,4) (3,5)	36
Nodos ↑	(1,2) (1,3) (2,4) (3,5)	36
Nodos ↓	(3,5) (3,4) (2,4) (1,3)	27
Nodos ↓	(3,5) (3,4) (2,4) (1,3)	27

Ilustración 5. Interfaz de usuario de TuMiST

TuMiST ofrece ocho funciones de selección:

- Peso de los arcos por orden creciente generando monografo
- Peso de los arcos por orden creciente generando multigrafo
- Peso de los arcos por orden decreciente generando monografo
- Peso de los arcos por orden decreciente generando multigrafo
- Peso de los nodos por orden creciente generando monografo
- Peso de los nodos por orden creciente generando multigrafo
- Peso de los nodos por orden decreciente generando monografo
- Peso de los nodos por orden decreciente generando multigrafo

El significado de multigrafo es la forma de indicar que se puede llegar a la solución a través de árboles independientes a partir del grafo original mientras que monografo se refiere a la forma de indicar que se puede llegar a la solución a través de un solo árbol que crece mientras avanza el algoritmo, hasta obtener el árbol de recubrimiento de todos los nodos.

La estrategia que corresponde al algoritmo de Kruskal es peso de los arcos por orden creciente generando multigrafo y la función de selección correspondiente al algoritmo de Prim es del peso de los arcos por orden creciente generando monografo.

TuMiST fue desarrollado inicialmente por los alumnos Ouafae Debdi y Juan David Granada. La mejora de los sistemas interactivos existentes era la primera etapa de este trabajo de tesis, con el objetivo de dar un mejor soporte a la docencia de algoritmos combinatorios, por lo cual se desarrolló una segunda versión de TuMiST.

En esta segunda versión, se eliminaron algunas deficiencias y se han adoptado algunas funciones de GreedEx al ayudante interactivo TuMiST como:

- La opción de la ejecución de un subconjunto de funciones de selección.
- La opción de la ejecución intensiva, que realiza una ejecución de un número muy alto de datos de entrada, donde el alumno tendrá la opción de elegir entre ejecutar sobre un número dado de datos de entrada o durante un determinado tiempo,
- Poder experimentar con TuMiST mientras se está realizando la ejecución intensiva.
- Cambio de la forma de posicionamiento de los nodos para mejorar la visualización.
- Adaptación de la aplicación al tamaño de la pantalla o resolución.
- Ampliación del tamaño de la ventana que contiene el grafo principal.
- Introducción de un JDialog para informar que el resultado ya está escrito en la tabla de resultados.
- La opción de ejecutar todas las estrategias.
- La opción de selección de estrategias desde la tabla de resultados.
- Añadir nuevas restricciones al ejecutar múltiples o todas las estrategias.

3.3 Sistema GreedEx

Con el objetivo de mejorar la enseñanza y el aprendizaje de los algoritmos voraces vía la herramienta educativa GreedEx (Velázquez-Iturbide, et al., 2010), se diseñó una versión más estable y mejorada de GreedEx, añadiendo nuevos problemas y mejorando las funciones existentes de la herramienta (Velázquez-Iturbide, et al., 2013a). Las sucesivas mejoras realizadas en GreedEx como

su extensión a nuevos problemas han sido desarrolladas mediante el uso combinado de prototipos y evaluaciones de usabilidad. Se realizaron dos clases de evaluación de usabilidad: de experto y de usuario. Las evaluaciones de experto fueron realizadas por el profesor de la asignatura donde se utilizó GreedEx.

GreedEx fue sometido a dos evaluaciones de usuario (detallados en el capítulo 4), donde la herramienta ha sido presentada a los alumnos, para evaluarla y opinar sobre ella (sugiriendo el añadir nuevas funciones o simplemente identificando los partes fáciles o difíciles en la misma), lo cual se identificaron nuevas mejoras a base de la opinión del usuario final, por lo que el proceso de la evaluación de experto anterior se repitió dos veces.

En este apartado se describen las distintas actividades realizadas para la mejora de GreedEx, salvo las evaluaciones de usabilidad con usuarios, que se describen con detalle en el Capítulo 4.

3.3.1 Estudio Bibliográfico

Se hizo una recopilación de numerosos problemas combinatorios que se pueden resolver mediante las técnicas de algoritmos voraces, algoritmos aproximados (o heurísticos) o programación dinámica (Debdi, et al., 2010). El objetivo de la recopilación de problemas efectuada ha sido estudiar las similitudes entre problemas y agrupar problemas relacionados, de forma que tengamos “familias” de problemas. Cada familia de problemas debe tener datos o funciones objetivo similares, aunque probablemente se resuelvan mediante técnicas de diseño distintas. La identificación de familias permitiría identificar familias de problemas a añadir a GreedEx. Al incluir problemas de una misma familia, se tendrían dos ventajas. Primero, añadir nuevos problemas con un esfuerzo pequeño, al reutilizar elementos entre problemas de la misma familia. Segundo, ilustrar cómo pequeñas variaciones en el enunciado de los problemas conducen a algoritmos óptimos basados en técnicas de diseño distintas.

La recopilación se ha realizado a partir de 14 libros de texto prestigiosos sobre algoritmos (Alsuwaiyel, 1999) (Baase & Gelder, 2000) (Brassard, 2001) (Cormen, et al., 2001) (Arroyo & Artacho, 2000) (Horowitz & Sahni, 1975) (Lee, et al., 2005)(Oliet, et al., 1999) (Neapolitan & Naimipour, 1999) (Kolikant, 2005) (Parberry, 1995) (Sahni, 1995) (Sedgewick, 2002) (Skiena, 2002). Esta revisión bibliográfica ha permitido caracterizar y clasificar diversos problemas de

planificación y posteriormente incorporar algunas variantes del problema de la mochila en el asistente interactivo GreedEx (la herramienta base del que se parte este trabajo).

La Tabla 7 presenta el resultado de analizar y agrupar problemas similares. Los problemas tienen planteamientos distintos: sobre tareas, objetos y contenedores, actividades, cintas y ficheros, programas y procesadores, etc. Para facilitar nuestra tarea, hemos reelaborado su planteamiento para que compartan un vocabulario común, el de tareas a realizar en cierto número de procesadores.

Las columnas de la Tabla 7 corresponden a cuatro grupos:

- “Identificador” del problema. Se ha tomado una numeración más su nombre según alguno de los libros donde aparece (sólo para los problemas que ya estaban implementados en AMO, SEDA, GreedEx o parecidos).
- Datos del problema. Hay una gran variedad de datos, que aparecen recogidos en las columnas 2-7: si las tareas tienen una duración individual, si tienen un tiempo de inicio y fin (caducidad), si tienen un plazo individual de realización (que sea ejecutada en un instante determinado), si tienen un beneficio individual asociado, si hay un plazo global, y el número de procesadores donde realizarlas. En cada columna aparece una X si dicho dato aparece en el enunciado del problema. Algunas columnas tienen un valor por defecto. La notación [X] en la columna de duración indica que este dato no aparece obligatoriamente en el enunciado del problema, porque puede deducirse de los datos inicio+fin, por ejemplo, en el enunciado viene que la tarea tiene una hora de comienzo y fin y esto nos indica que tiene una duración aunque no está escrita en el enunciado.
- Función objetivo del problema. La función objetivo consta de tres partes: maximización o minimización, la “relación combinatoria” de los datos seleccionados con los datos de entrada (si es un subconjunto, si es una permutación, si hay que partirlos en varios conjuntos), y cuál es la función objetivo a optimizar. Esta última puede tomar la forma de hallar un número (#), una suma o un máximo. Asimismo, puede calcularse sobre las tareas (T), su duración (D, quizá acumulando tiempos de espera), su beneficio (B)) y el número de procesadores (P).

- Técnica de diseño con la que el problema se resuelve en la bibliografía: algoritmo voraz (V), programación dinámica (PD) o algoritmo aproximado (A).

El resultado son 7 grupos de problemas. Puede observarse la similitud de los problemas de cada grupo.

Tabla 7. Agrupación de problemas

Nombre	Duración ¹	Inicio+fin	Plazo indiv.	Beneficio	Plazo global	# Proces. ²	MAX/mi subconj., ordenar, repartir ³	datos	Técnica de diseño
Problema 2. Maximizar el número de cintas	X				X		MAX subconj.	# T	¿V?
Problema 1. Maximum programs stored problem	X				X	2	MAX subconj.	# T	A
Problema 3. Maximizar el espacio utilizado del disco	X				X		MAX subconj.	suma D	¿V?
Problema 4. Mochila	X		X		X		MAX subconj.	suma B	PD
Problema 5. Mochila 0/1	X		X		X		MAX subconj. con frac.	suma B	V
Problema 7. n tareas en dos procesadores	X^4					2	min repartir	MAX (2 sumas)	PD
Problema 8.	X					X	min repartir	Max D	A
Problema 6. Almacenar n programas en dos cintas	X				X	2	min repartir	MAX (2 sumas)	PD
Problema 9.	X				X		min repartir	# P	A
Problema 10.	[X]	X					min repartir	# P	V
Problema 11. Planificación con plazo fijo			X	X			MAX subconj. ordenado	suma B	V
Problema 12.	X		X	X			MAX subconj. ordenado	suma B	V
Problema 13. Selección de actividades	[X]	X					MAX subconj.	# T	V
Problema 15.	X						min ordenar	suma D acumul.	V
Problema 14.	X^5						min ordenar	suma D acumul.	V

¹ Por defecto, tienen duración unitaria.

² Por defecto, es un procesador.

³ Repartir entre dos procesadores es equivalente a determinar un subconjunto a ejecutar en un solo procesador.

⁴ Duración variable para cada procesador.

Problema 16.	X	Inicio	min	ordenar con fracciones	suma D acumul.	V	
Problema 17.	X		X	min	ordenar	suma D acumul.	V

3.3.2 Nuevos Problemas Soportados

La revisión bibliográfica descrita en la sección anterior permitió caracterizar y clasificar diversos problemas de planificación y posteriormente incorporar algunas variantes del problema de la mochila en el asistente interactivo GreedEx. Los otros problemas se dejaron para nuevas ampliaciones. En esta sección se detallan los nuevos problemas (tipo mochila) soportados por GreedEx.

3.3.2.1 Problema de Maximizar el Número de Objetos

GreedEx ha sido ampliado con el problema de Maximizar Número de Objetos(Oliet, et al., 1999) (problema 2 en la Tabla 7). El planteamiento del problema es el siguiente:

Se tienen n objetos, cada uno con un peso $p_s[i]$, $0 \leq i \leq n-1$, y una mochila con una capacidad c . El objetivo es maximizar el número de objetos que se introducen en la mochila sin sobrepasar la capacidad de ésta. Para que el problema no tenga una solución trivial (introducir todos los objetos), la suma de todos los pesos debe ser mayor que la capacidad de la mochila. Por ejemplo, sea el siguiente caso, con una mochila $c = 15$ y tres objetos con $p_s = \{9,6,5\}$. Cualquier solución que permita introducir dos objetos en la mochila es óptima. Las funciones de selección integradas en GreedEx para este problema son:

- Orden creciente de índice
- Orden decreciente de índice
- Orden creciente de peso
- Orden decreciente de peso

⁵ Cada tarea tiene, además, probabilidad de petición.

3.3.2.2 Problema de Maximizar el Peso de Objetos

Otro problema tipo mochila que ha sido añadido en el asistente interactivo GreedEx, es el de Maximizar el Peso de Objetos (Oliet, et al., 1999) (problema 3 en la Tabla 7). El planteamiento del problema es el siguiente:

Se tienen n objetos, cada uno con un peso $p_s[i]$, $0 \leq i \leq n-1$, y una mochila con una capacidad c . Un objeto i puede introducirse o no en la mochila. La decisión de introducir o no un objeto en la mochila se denota $x_s[i]$, $x_s[i] \in \{0,1\}$. El objetivo es maximizar el peso introducido en la mochila, es decir:

$$\text{maximizar } \sum_{i=0}^{n-1} p_s[i] \cdot x_s[i].$$

Para que el problema no tenga una solución trivial ($x_s[i]=1$, $0 \leq i \leq n-1$), la suma de todos los pesos debe ser mayor que la capacidad de la mochila. Por ejemplo, sea el siguiente caso, con una mochila $c = 15$ y tres objetos con $p_s = \{9,6,5\}$. Veamos dos soluciones posibles en la Tabla 8 donde la solución óptima es la última.

Tabla 8. Posibles soluciones de el problema de maximizar el peso de objetos en una mochila

$\{x_s[0], x_s[1], x_s[2]\}$	$\sum p_s[i] \cdot x_s[i]$
{0,1,1}	11
{1,1,0}	15

Las funciones de selección integradas en el sistema GreedEx para este problema son:

- Orden creciente de índice
- Orden decreciente de índice
- Orden creciente de peso
- Orden decreciente de peso

3.3.2.3 Problema de Maximizar el Número de Objetos en dos Mochilas

El último problema que ha sido añadido en GreedEx es el de Maximizar el Número de Objetos en Dos Mochilas (Horowitz & Sahni, 1975) (problema 1 en la Tabla 7). El planteamiento del problema es el siguiente:

Se tienen n objetos, cada uno con un peso $p_s[i]$, $0 \leq i \leq n-1$, y dos mochilas con capacidades c_1 y c_2 . El objetivo es maximizar el número de objetos que se introducen en ambas mochilas sin sobrepasar sus capacidades. Para que el problema no tenga una solución trivial (introducir todos los objetos), la suma de todos los pesos debe ser mayor que las capacidades de las mochilas. Por ejemplo, sean cinco objetos con $p_s = \{9, 6, 5, 7, 4\}$ y las dos mochilas con $c_1=15$ y $c_2=16$. Una solución óptima es introducir los objetos de pesos 9,6 en la primera mochila y los de pesos 5, 7, 4 en la segunda. La solución consistente en introducir los objetos de pesos 9,5 en la primera mochila y los de pesos 6,7 en la segunda es válida pero subóptima. Las funciones de selección integradas en GreedEx de este problema son:

- Orden creciente de índice, mochila más vacía
- Orden decreciente de índice, mochila más vacía
- Orden creciente de peso, mochila más vacía
- Orden decreciente de peso, mochila más vacía
- Orden creciente de índice, mochila más llena
- Orden decreciente de índice, mochila más llena
- Orden creciente de peso, mochila más llena
- Orden decreciente de peso, mochila más llena

3.3.2.4 Problema de la Mochila 0/1

La ampliación de GreedEx con el problema de la mochila 0/1 ha enriquecido el asistente, dado que este problema no se solventa usando un algoritmo voraz, sino un algoritmo aproximado (problema 5 en la Tabla 7). El planteamiento del problema de la mochila 0/1 es el siguiente:

Se tienen n objetos, cada uno con un peso $p_s[i]$, $0 \leq i \leq n-1$, y una mochila con una capacidad c . Un objeto i puede introducirse o no en la mochila; en el primer caso, se obtiene un beneficio $b_s[i]$, mientras que en el segundo caso, no se obtiene ningún beneficio. La decisión de introducir o no un objeto en la mochila se denota $x_s[i]$, $x_s[i] \in \{0,1\}$. El objetivo es llenar la mochila de forma que se

maximice el beneficio total, es decir: $\max imizar \sum_{i=0}^{n-1} b_s[i] \cdot x_s[i]$.

Para que el problema no tenga una solución trivial ($x_s[i]=1, 0 \leq i \leq n-1$), la suma de todos los pesos debe ser mayor que la capacidad de la mochila. Por ejemplo, sea el siguiente caso, con una mochila $c = 15$ y tres objetos con $p_s = \{9,6,5\}$ y $b_s = \{38,40,24\}$. Veamos en la Tabla 9 dos soluciones posibles, donde la solución óptima es la última.

Tabla 9. Posibles soluciones Mochila 0/1

$\{x_s[0], x_s[1], x_s[2]\}$	$\sum b_s[i] \cdot x_s[i]$
{0,1,1}	64
{1,1,0}	78

Se han integrado las siguientes funciones de selección para este problema en GreedEx:

- Orden creciente/decreciente de índice.
- Orden creciente/decreciente de peso.
- Orden creciente/decreciente de beneficio.
- Orden creciente/decreciente de beneficio/peso.
- Orden creciente/decreciente de peso/beneficio.

3.3.3 Evolución de GreedEx

GreedEx ha sido evaluado constantemente, haciéndolo más sostenible y fácil de usar (Velázquez-Iturbide, et al., 2013a) para soportar mejor la tarea educativa prevista.

La Tabla 10 muestra las mejoras realizadas en GreedEx a base de dos evaluaciones de usabilidad que describimos en detalle en el próximo capítulo y su desarrollo en cada evaluación. Las celdas marcadas con una X significan que el sistema en la correspondiente columna soporta la función en la correspondiente fila.

Los cambios principales de GreedEx han sido la movilidad del panel de visualización, exportación de tablas, mejorar de la ejecución intensiva, añadir las restricciones de la interfaz de usuario sobre las distintas granularidades de experimentación, añadir los logs, incluir la ayuda interactiva y añadir los problemas tipo mochila.

En resumen, tras la primera evaluación de usabilidad han sido agregados varias funcionalidades como por ejemplo la exportación de tablas, ayuda interactiva y ejecución guiada y se mejoró la ejecución intensiva y los iconos. Tras la segunda evaluación, varios elementos de GreedEx han sido revisados como los iconos, funciones de exportación, teclas de atajo y creación de fichero logs. Además se agregaron otros problemas variantes del problema de la mochila y otras funcionalidades como la exportación de tablas en formato Excel y la internalización de la ayuda etc.

Tabla 10. Mejoras de GreedEx

CARACTERÍSTICAS	EVA. 1	EVA. 2
Movilidad del panel de visualización	X	
Exportación de tablas	X	
Integrar los índices en la visualización para indicar el orden de selección	X	
Mejora de la ejecución intensiva*	X	
Restricciones de la interfaz de usuario sobre las distintas granularidades de experimentación	X	
Añadir los logs	X	
Ayuda interactiva	X	
Añadir la opción de la ejecución guiada	X	
Mejorar los Iconos	X	X
Revisión de teclas de atajo		X
Revisión de las funciones de exportación de tablas		X
Revisión de la creación de ficheros de logs		X
Añadir el problema de “Maximizar el número de objetos”		X
Añadir el problema de “Maximizar el número de objetos en Dos Mochilas”		X
Añadir el problema de la “Mochila 0/1		X
Añadir el problema de “Maximizar el peso de los objetos”		X
Traducir el resto del sistema al ingles		X
Internalización de la ayuda		X
Exportación de tablas en Excel		X
*Dar mejor soporte para la funcionalidad de la ejecución intensiva, incorporando un número grande de ejecuciones sin que la aplicación se quede bloqueada, permitiendo al usuario realizar otras acciones que no interfieran con la de ejecución intensiva y añadiendo porcentajes de optimalidad para cada función de selección.		

Por otro lado, GreedEx tiene una ayuda interactiva disponible en el menú principal de la herramienta. En esta ventana se puede encontrar toda la información relativa al uso y el funcionamiento de GreedEx mediante la navegación por las diferentes secciones que permite pasar de un tipo de información a otra a través de vínculos o enlaces. Nuestra herramienta está disponible en la página Web del departamento en el siguiente enlace <http://www.lite.etsii.urjc.es/GreedEx/>, para su descarga y para obtener más información sobre ella. Por otro parte, GreedEx se ofrece en dos idiomas, en español y en inglés, para lograr su internacionalización y poderla usar en otros países. Es más, GreedEx permite exportar los resultados de la experimentación, ya sea como una visualización individual o un conjunto de visualizaciones. Se almacenan en formatos de archivo estándar que luego pueden ser importados en los documentos, por ejemplo, notas de texto o informes de laboratorio.

3.3.4 Hallazgos Experimentales

La experimentación con optimalidad desempeña un papel central en el aprendizaje de algoritmos voraces (Velázquez-Iturbide & Debdi, 2011). El método experimental propuesto permite a los alumnos experimentar con diferentes funciones de selección para un problema de optimización dado. Después se les pide razonar con rigor para determinar las funciones de selección óptimas, justificando sus hallazgos experimentales usando la herramienta educativa GreedEx. El alumno debe saber cómo planificar una experimentación a partir de los resultados obtenidos, descartando la función de selección que no ha obtenido un resultado óptimo utilizando los datos de entrada empleados como un contraejemplo para demostrar la no optimalidad de ésta.

En este apartado examinamos unos resultados experimentales obtenidos con los distintos problemas integrados en GreedEx. Algunos problemas pueden resolverse de forma óptima con la técnica voraz mientras que otros no; en este último caso, podemos clasificar las funciones de selección subóptimas como heurísticas mejores o peores. También hay que avisar que no conocíamos a priori los resultados a obtener, por lo que algunos nos han resultado sorprendentes.

Una consecuencia de la propuesta de un conjunto de funciones de selección diferentes es que ilustran que un problema dado puede ser resuelto de manera óptima con varias funciones de selección. Considere el problema de la mochila (mencionada arriba en la sección 3.2.1.2). La Tabla 11 muestra el porcentaje de optimalidad de las funciones de selección propuestas tras realizar 10.000

ejecuciones. En este problema, podemos ver que hay funciones de selección equivalentes por ejemplo la función de selección de orden decreciente de beneficio/peso es óptima, pero también lo es la función de selección orden creciente de peso/beneficio. Es decir, las dos funciones de selección producen el mismo orden de elementos o fracciones de elementos.

Tabla 11. Mochila

Mochila	
Función de selección	Porcentaje
Orden creciente de índice	8.03
Orden decreciente de índice	7.9
Orden creciente de peso	28.17
Orden decreciente de peso	1.5%
Orden creciente de beneficio	1.13
Orden decreciente de beneficio	33.59
Orden creciente de peso/beneficio	100
Orden decreciente de peso/beneficio	0.0
Orden creciente de beneficio/peso	0.0
Orden decreciente de beneficio/peso	100
Se han realizado 10.000 ejecuciones	

Otro problema similar al de la mochila pero más sencillo es el problema de maximizar el número de objetos en una mochila (el enunciado se encuentra en la sección 3.3.2.1). En este problema la función de selección óptima es el de orden creciente de peso. Este experimento proporciona evidencia de la optimalidad de esta función de selección aunque se debe demostrar formalmente su optimalidad (Oliet, et al., 1999).

Tabla 12. Maximizar el número de objetos

Maximizar el número de objetos	
Función de selección	Porcentaje
Orden creciente de índice	70.38
Orden decreciente de índice	70.24
Orden creciente de peso	100
Orden decreciente de peso	38.6
Se han realizado 10.000 ejecuciones	

El método didáctico no sólo promueve el rigor sino también la creatividad dado que para algunos problemas se puede encontrar funciones de selección óptimas y casi óptimas. Veámoslo con el problema de selección de actividades (sección 3.1.1). En la Tabla 13 se puede ver que las dos funciones de selección óptimas son orden decreciente de comienzo y orden creciente de fin (funciones de selección equivalentes). Pero además, nos encontramos con funciones de selección casi óptimas. Por ejemplo, la función de selección “orden creciente de duración” es óptima para un alto porcentaje de los casos (97.77%). En este caso, el alumno debe encontrar un contraejemplo para demostrar la no optimalidad de esta función de selección.

Tabla 13. Selección de actividades

Selección de actividades	
Función de selección	Porcentaje
Orden creciente de índice	63.84
Orden decreciente de índice	62.89
Orden creciente de comienzo	52.9
Orden decreciente de comienzo	100
Orden creciente de fin	100
Orden decreciente de fin	52.73
Orden decreciente de duración	97.77
Orden creciente de duración	24.37
Se han realizado 10.000 ejecuciones	

La experimentación también se puede aplicar a los problemas para los que ninguna función de selección es óptima. En este caso, el proceso de experimentación sirve para descartar los algoritmos voraces como una técnica de diseño adecuada para resolver el problema de manera óptima. En consecuencia, otras técnicas de diseño deben ser considerados. La Tabla 14 muestra el porcentaje de optimalidad de las funciones de selección para el problema de la mochila 0/1 (sección 3.3.2.4), según esta tabla ninguna función de selección es óptima.

Tabla 14. Mochila 0/1

Mochila 0/1	
Función de selección	Porcentaje
Orden creciente de índice	30.48
Orden decreciente de índice	31.43
Orden creciente de peso	37.34
Orden decreciente de peso	22.36

Orden creciente de beneficio	16.02
Orden decreciente de beneficio	84.37
Orden creciente de peso/beneficio	81.16
Orden decreciente de peso/beneficio	15.78
Orden creciente de beneficio/peso	15.78
Orden decreciente de beneficio/peso	81.16
Se han realizado 10.000 ejecuciones	

Otros problemas son sencillos y sin embargo no se pueden resolver mediante un algoritmo voraz como es el caso del problema de maximizar el peso de objetos en una mochila (sección 3.3.2.2). Según la Tabla 15, se puede observar que ninguna función de selección es óptima. Este problema puede resolverse mediante programación dinámica, vuelta atrás y ramificación y poda (Oliet, et al., 1999).

Tabla 15. Maximizar el peso de objetos en una mochila

Maximizar el peso de objetos en una mochila	
Función de selección	Porcentaje
Orden creciente de índice	37.56
Orden decreciente de índice	37.63
Orden creciente de peso	19.93
Orden decreciente de peso	77.01
Se han realizado 10.000 ejecuciones	

En el problema de maximizar el número de objetos en dos mochilas (sección 3.3.2.3) ninguna de las funciones de selección es óptima según la Tabla 16. Obsérvese que la función de selección "orden creciente de peso, mochila más vacía" es óptima en el 90% de los casos. Por tanto, es una buena heurística, aunque faltaría determinar su desviación relativa respecto a la solución óptima para saber si puede ser un buen algoritmo aproximado (Horowitz & Sahni, 1975).

Tabla 16. Maximizar el número de objetos en dos mochilas

Maximizar el número de objetos en dos mochilas	
Función de selección	Porcentaje
Orden creciente de índice, mochila más vacía	63.99
Orden decreciente de índice, mochila más vacía	64.74
Orden creciente de peso, mochila más vacía	73.08
Orden decreciente de peso, mochila más vacía	40.4
Orden creciente de índice, mochila más llena	69.7

Orden decreciente de índice, mochila más llena	70.29
Orden creciente de peso, mochila más llena	90.76
Orden decreciente de peso, mochila más llena	41.19
Se han realizado 10.000 ejecuciones	

En resumen, el método experimental fue concebido para hacer frente a los algoritmos voraces, aunque también se puede aplicar para otras técnicas de diseño de algoritmos: programación dinámica, vuelta atrás, *branch and bound* y los algoritmos de aproximación, lo que convierte al sistema GreedEx a un instrumento para la introducción a otros algoritmos.

3.4 Ampliación de GreedEx

En esta sección se describe la herramienta GreedExCol (Velázquez-Iturbide, et al., 2013c), su interfaz y las tecnologías de desarrollo usadas.

3.4.1 Experimentación Colaborativa

GreedEx ha sido ampliado para apoyar la discusión colaborativa como último paso en el método experimental. La incorporación de este enfoque colaborativo con la ayuda del ordenador (CSCL) mejora la calidad del aprendizaje desarrollando habilidades de pensamiento de alto nivel (Gokhale, 1995) y mejora del rendimiento académico (Prince, 2004). Este enfoque permite soportar un espacio colaborativo para compartir los experimentos realizado por un grupo de alumnos, concretamente facilitando un espacio para propuestas y contribuciones en grupo.

GreedExCol es una herramienta interactiva e innovadora que junta las ventajas del aprendizaje activo mediante experimentación y colaboración para impulsar el aprendizaje de los algoritmos voraces.

Sistema GreedExCol

En esta sección se describe la ampliación colaborativa GreedExCol, su interfaz gráfica y las tecnologías usadas.

3.4.2.1 Descripción de GreedExCol

GreedExCol soporta un espacio de contribuciones que ayuda al debate (o “discusión”) y toma de decisiones *face-to-face* dentro de un grupo de alumnos sobre la solución de un problema. Por tanto, se supone que cada alumno trabaja de forma individual en las primeras etapas de experimentación (enfoque Cognitivo del Constructivismo promulgado por Piaget) y posteriormente interviene en un debate con el resto de los miembros del grupo para decidir qué funciones de selección son óptimas a partir de los datos obtenidos por todos ellos (enfoque Socio-cognitivo del Constructivismo promulgado por los conocidos neo-piagetianos o escuela de Génova). Para realizar este debate, los miembros de un grupo comparten los datos obtenidos experimentalmente, mostrados en las visualizaciones y en las tablas.

El sistema supone que los alumnos están organizados en pequeños grupos y comparten los resultados de sus experimentos. Un proceso de trabajo corriente para una sesión de prácticas es el siguiente:

- Se forman grupos de un máximo de 4 alumnos. No hay ninguna asignación de roles.
- El profesor explica el objetivo de la práctica (identificar las funciones de selección óptimas) y presenta el informe que los alumnos tienen que entregar.
- Los alumnos se registran y comienzan a experimentar individualmente con GreedExCol, reflexionando sobre los resultados obtenidos para el problema a resolver. La herramienta soporta un espacio personal e individual para cada miembro del grupo.
- Se comparten resultados y conclusiones. Es en esta etapa del proceso donde GreedExCol aporta una innovación en el aprendizaje de los algoritmos voraces, proporcionando dos tipos de espacios: un espacio de comunicación donde se produce el debate (compartiendo los datos obtenidos en el grupo) y un espacio de trabajo donde cada alumno trata de llegar a una conclusión por su cuenta.
- Los alumnos escriben el informe que describe sus hallazgos y lo envían al profesor.
- El profesor revisa las soluciones, señala errores y otras cuestiones interesantes para consolidar conocimientos.

GreedExCol proporciona dos espacios de trabajo que gestionan estas interacciones sociales y las reflexiones individuales:

- Espacio individual. El alumno experimenta con GreedExCol de la misma manera que con GreedEx. La única diferencia consiste en que los datos y resultados se guardan en una base de datos centralizada para poder compartirlos. De hecho, cuando el alumno vuelve a iniciar su sesión como usuario del sistema, todos los experimentos anteriormente realizados se cargan y se ejecutan automáticamente. Este espacio individual facilita la reflexión personal.
- Espacio grupal. El alumno puede ver simultáneamente los experimentos de todo el grupo o solo los de algún otro miembro. Esta información puede actualizarse a demanda del interesado, por si sus compañeros han realizado más actividades individuales. También puede realizar nuevas ejecuciones en el espacio de trabajo de un compañero, que se guardarán y se enviarán automáticamente como propuesta a dicho compañero. Cada propuesta se acompaña de un mensaje y puede ser aceptada o rechazada. Este espacio grupal facilita la interacción social.

3.4.2.2 Interfaz de GreedExCol

Para empezar a experimentar con GreedExCol, el alumno debe registrarse en el sistema para tener un usuario y una clave. Durante el registro, el alumno debe rellenar varios campos: nombre del usuario, contraseña, pregunta de seguridad, dirección de correo electrónico y grupo. En este último campo se muestra la lista de grupos formados (el profesor asigna verbalmente a cada alumno un grupo). El número de grupo de un usuario aparecerá siempre en la parte derecha de la pantalla (véanse Ilustración 6, etiquetas Grupo). El número máximo de alumnos por grupo es cuatro.

Un alumno puede entrar en el espacio grupal al seleccionar la opción de menú “Discusión” y regresar a su espacio individual pulsando la opción de menú “Cerrar vista discusión”. También puede regresar sin salir del espacio de propuestas, seleccionando la pestaña que lleva su nombre en la barra de botones (véase Ilustración 6, etiqueta C).

En la Ilustración 6, vemos el espacio grupal junto con una nueva barra de botones que muestra:

1. El botón “Comparativa” permite mostrar simultáneamente los espacios de todos los miembros del grupo (véase Ilustración 6, etiqueta A).
2. Hay tantos botones como número de miembros que en el grupo. Al seleccionar uno de estos botones, se muestra el espacio individual del compañero correspondiente (véase Ilustración 6, etiqueta C).
3. Actualizar el trabajo de los compañeros. Bajo demanda, se actualizan los datos de todo el grupo, es decir, sus últimas ejecuciones, tablas y propuestas (véase Ilustración 6, etiqueta D).
4. Propuesta. GreedExCol permite a los alumnos utilizar el envío de propuestas como método de comunicación entre los miembros del grupo. Una propuesta es un experimento formado por los datos de una o varias ejecuciones de un compañero. Cada propuesta se acompaña de un mensaje (véase Ilustración 7, etiqueta C) para que el receptor de la propuesta pueda saber qué quiere decir su compañero con el experimento enviado. Al seleccionar la propuesta de un compañero del grupo (véase Ilustración 7, etiqueta B), la pantalla muestra solamente su espacio de ejecución y aparece una nueva barra de botones (véase Ilustración 7). La nueva barra permite aceptar la propuesta (Ilustración 7, etiqueta A), rechazarla (Ilustración 7, etiqueta B) y ver el mensaje que la acompaña (Ilustración 7, etiqueta C). Al rechazarla se elimina definitivamente y si se acepta se almacena permanentemente en el espacio individual del alumno.

Es más, la herramienta proporciona algunas funciones específicas para el profesor:

- Crear grupos de alumnos para las sesiones de trabajo y asignar el número de grupo a cada alumno.
- Ver el número de alumnos registrados en GreedExCol.
- Ver los grupos creados. Permite ver la lista de alumnos de cada grupo.
- Cambiar el grupo de cualquier alumno registrado.
- Cambiar el nombre de usuario del alumno (Nick).
- Vaciar la base de datos. El profesor podrá eliminar todos los usuarios y todos los experimentos realizados de los problemas de GreedExCol.

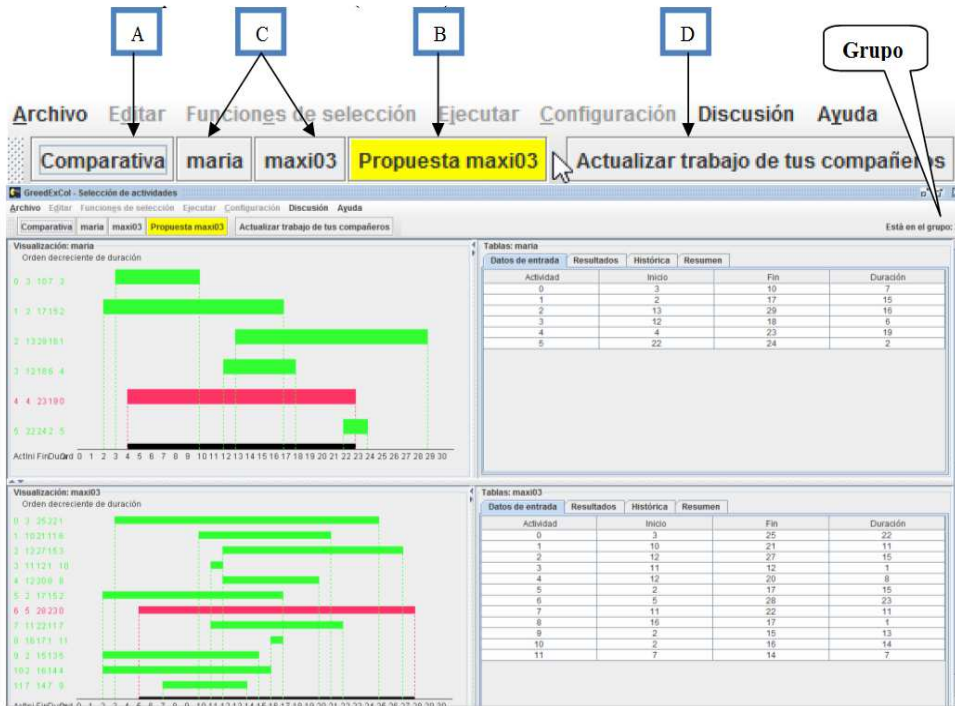


Ilustración 6. Aspecto de GreedExCol para un grupo de dos alumnos

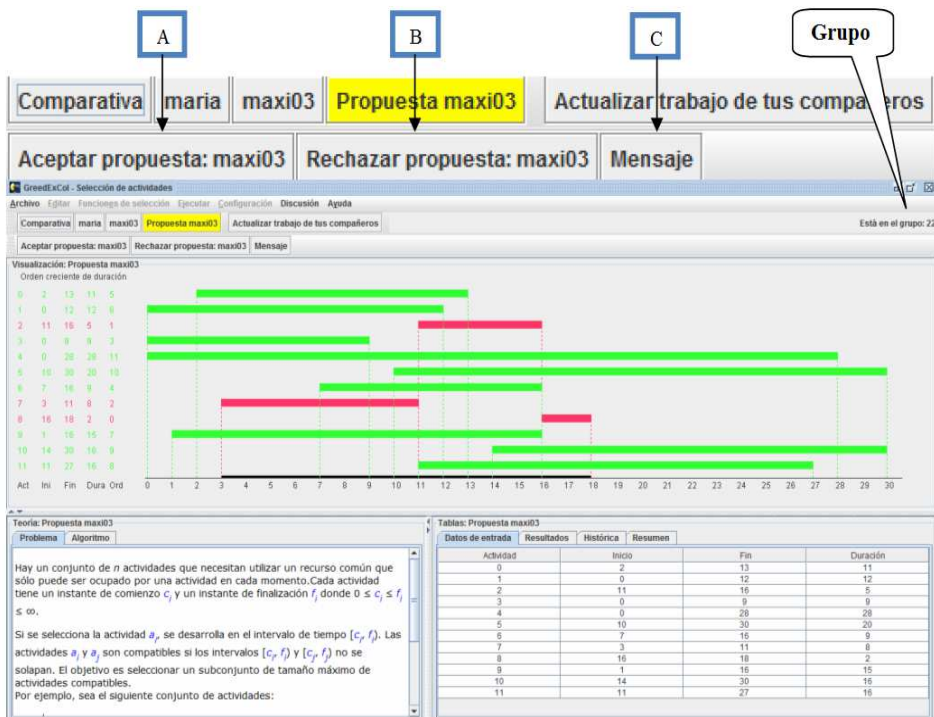


Ilustración 7. Aspecto de GreedExCol para aceptar/rechazar propuesta

3.4.3 Tecnologías de Desarrollo

Tanto el desarrollo de mejoras, nuevas funcionalidades y nuevos problemas añadidos en GreedEx como el desarrollo de su extensión colaborativa GreedExCol, han sido implementados en Java usando Eclipse como IDE de desarrollo.

GreedExCol se basa en una arquitectura centralizada bajo un modelo cliente-servidor. En este modelo, el cliente envía un mensaje solicitando un servicio en particular al servidor y éste envía uno o más mensajes de respuesta. El servidor es el proceso responsable de servir a varios clientes que realizan solicitudes de un recurso administrado por él. El proceso del servidor se conoce con el término de “*back-end*”. El cliente es el proceso que permite al usuario formular los requisitos y pasarlos al servidor, por eso se conoce con el término “*front-end*”.

En el caso de GreedExCol, el alumno crea su cuenta rellenando los campos necesarios y comienza a experimentar, el alumno puede volver a su cuenta con su identificación personal, el servidor envía esta información a la base de datos y carga sus experimentos realizados anteriormente. Por otro lado, cualquier alumno puede enviar una propuesta para cualquier miembro de su grupo. Como se muestra en la Ilustración 8, El alumno 4 envía una propuesta para el alumno 1, entonces el servidor la envía a la base de datos. Cuando el alumno 1 actualiza su interfaz, puede notar la nueva propuesta presentada por el alumno 4. La Ilustración 8 muestra la arquitectura de red.

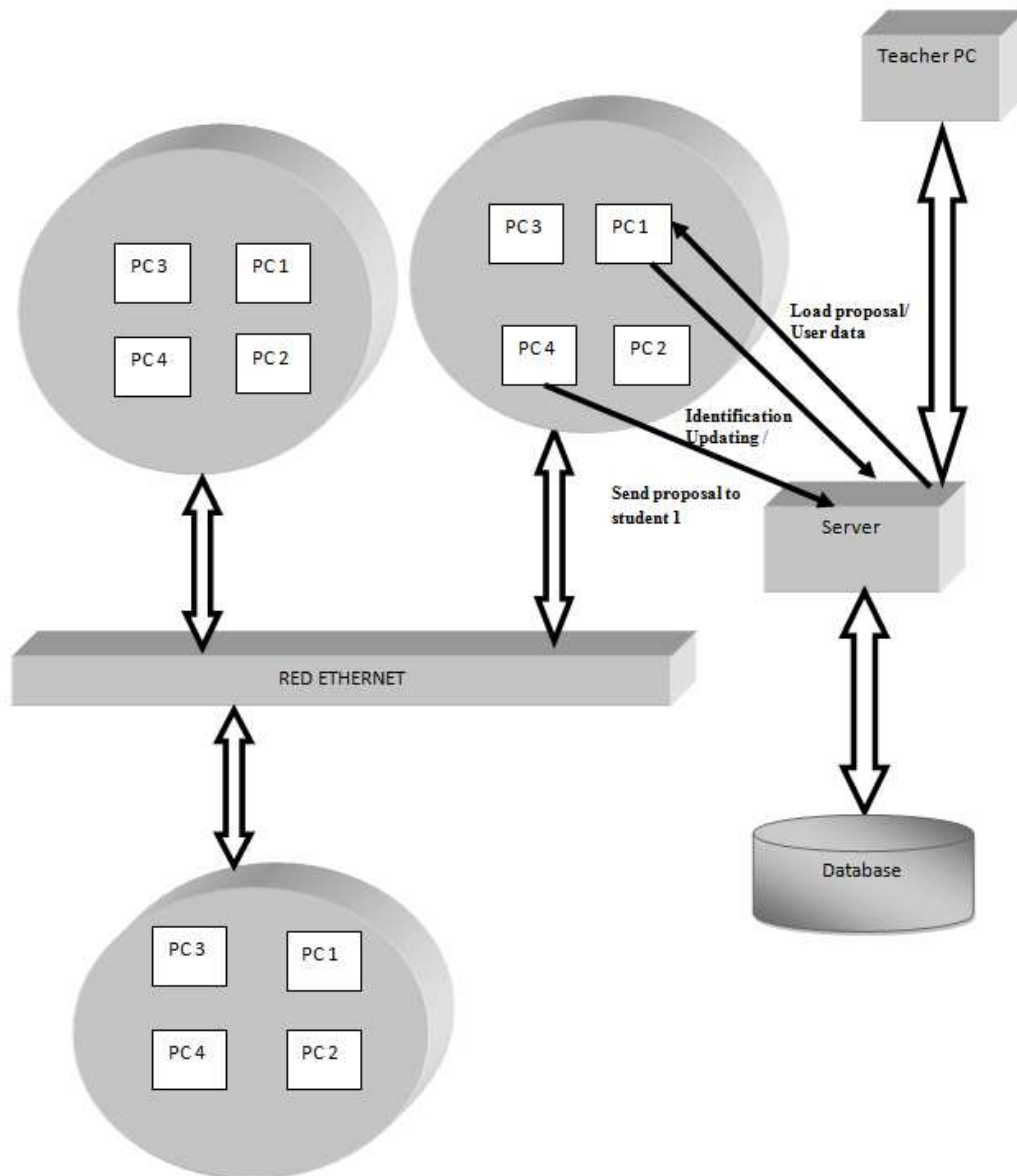


Ilustración 8. Arquitectura de red

Los experimentos realizados por el alumno se almacenan en una base de datos utilizando MySQL (usando PHPMyAdmin para la gestión de la base de datos) que se utiliza para añadir, eliminar y modificar la información en la base de datos. MySQL proporciona conectividad a la base de datos para las aplicaciones cliente desarrolladas en el lenguaje de programación Java a través de un controlador JDBC (MySQL-Connector para Java). El controlador JDBC se utiliza para conectarse a la base de datos. El controlador JDBC que reside en el sistema local se conecta a la base de datos y

puede acceder a la base de datos a través de la red. Esta configuración es de tipo Cliente/Servidor donde el programa cliente envía instrucciones SQL a la base de datos; los procesa y envía los resultados a la aplicación como puede verse en la Ilustración 9.

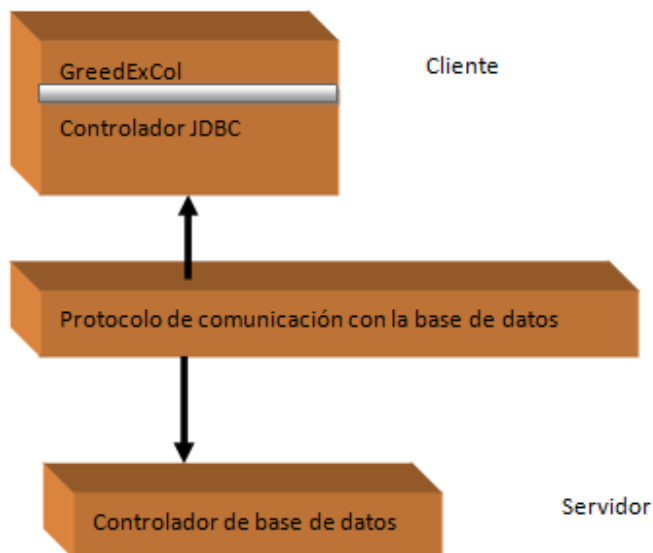


Ilustración 9. Acceso de JDBC a la base de datos

La base de datos de GreedExCol cuenta con ocho tablas. Seis tablas para cada problema integrada en GreedExCol (mochila, mochila 0/1...). En cada una de estas tablas se almacena los experimentos realizados por el alumno y que tienen los mismos campos: "codigo" (que se genera automáticamente), "datos" (datos de entrada del problema), "estrategias" (funciones de selección ejecutados), "usuario" (nombre del usuario), "propuestoPor" (el nombre del usuario que propuso el experimento o el propio usuario si no es una propuesta). Las dos restantes tablas son: " propuesta" que contiene información relacionada con la propuesta incluyendo los mensajes acompañados y la tabla "usuarios" que contiene la información sobre el usuario (nombre de usuario, contraseña, pregunta de seguridad, correo electrónico, grupo e IP). La Ilustración 10 muestra la base de datos de GreedExCol.

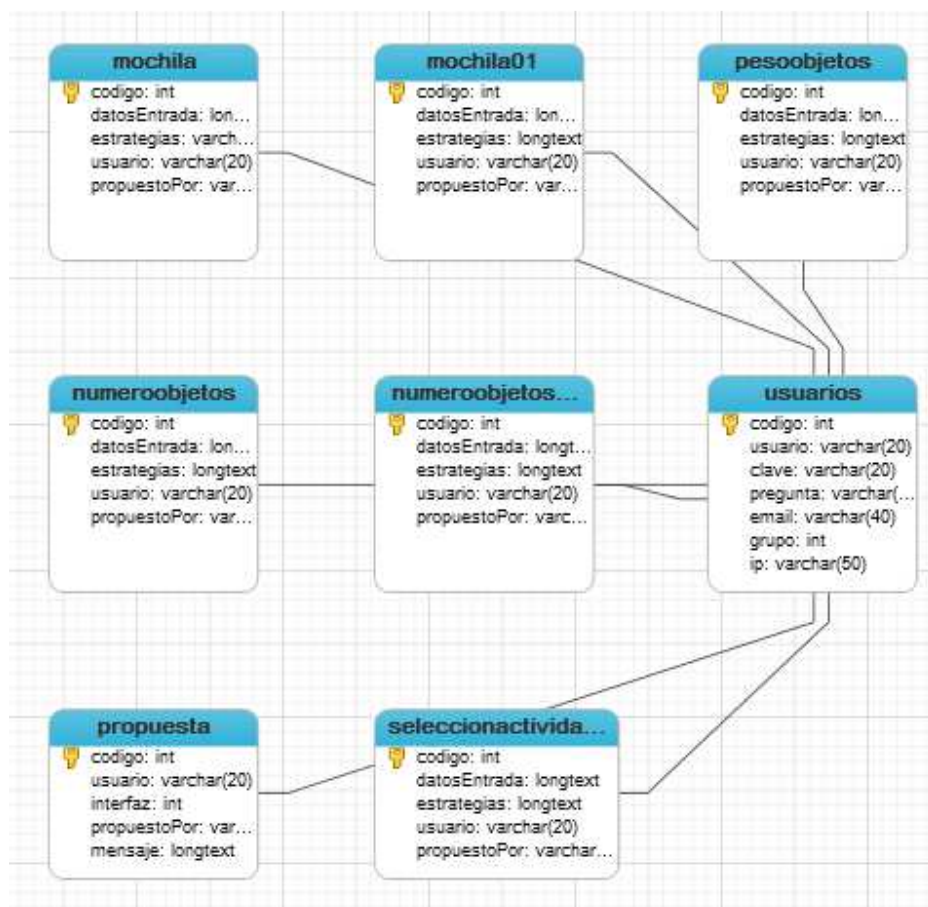


Ilustración 10. Base de datos de GreedExCol

3.4.3 Arquitectura de software

La arquitectura de software de GreedExCol sigue el patrón Modelo-Vista-Controlador (MVC) (WikiBooks, 2014)(Gamma, et al., 1994). GreedEx se compone de varios problemas por esta razón el patrón MVC es ideal e útil dado que separa la interfaz del modelo y permite que se comuniquen a través del controlador. La arquitectura de GreedExCol permite añadir nuevos problemas. Cuando el usuario solicita una opción se le notifica a la parte del programa que se encarga del control, y este le encarga la ejecución al modelo. El modelo es distinto para cada programa y puede ser desarrollado independientemente de las partes de la vista y el control que son comunes a todos.

En GreedExCol, hay tres paquetes " Interfaz", "FicherosOpciones" y "FiltrosFicheros" que forman la vista que es la representación del modelo o la interfaz de usuario. El paquete "Control" es el controlador que responde a los eventos, por lo general las acciones del usuario e invoca cambios en

el modelo. Y el paquete "ManipulacionDatos" es el modelo que encapsula los datos y la funcionalidad. Es independiente de cualquier representación de la producción y/o el comportamiento de entrada. Y por último el ControladorD es la responsable de la lectura y control de acceso a la base de datos.

Interfaz. Crea la ventana que forma la interfaz gráfica de usuario. Muestra y solicita información al usuario.

Control. Contiene la clase "Acciones" que controla todo lo que sucede en la aplicación, entre otras clases.

FicheroOpciones. Contiene clases responsables de la manipulación de los archivos involucrados para la configuración de la aplicación. Sirve como un mecanismo para la intercomunicación entre las clases. También contiene todas las clases que simplemente construyen una estructura de datos, que consta de atributos y métodos para obtener su valor (*get*) y modificar los (*set*).

FiltroFicheros. Cuenta con seis clases (FiltroFicheroTexto, FiltroXML, FiltroJPG, FiltroPDF, FiltroXLS y FiltroGIF) que desarrollan la clase abstracta de FileFilter.

ManipulacionDatos. Contiene clases responsables de la aplicación de algunas de las funcionalidades que proporciona la aplicación.

ControladoBD. Contiene todos los métodos responsables de enviar comandos a la base de datos. Es responsable de la recogida y la inserción de datos en la base de datos.

Cuando el usuario interactúa con la interfaz, la vista envía al controlador que recibe la acción solicitada por el usuario y gestiona el evento necesario para que se realice esta petición. Para ello accede al modelo, modificándolo de forma adecuada a la acción solicitada por el usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada. De esta forma, todas las funcionalidades que se encuentran en la interfaz de usuario se pueden modificar o añadir otras sin que repercuta en otras partes del código. La Ilustración 11 muestra los paquetes de GreedExCol.

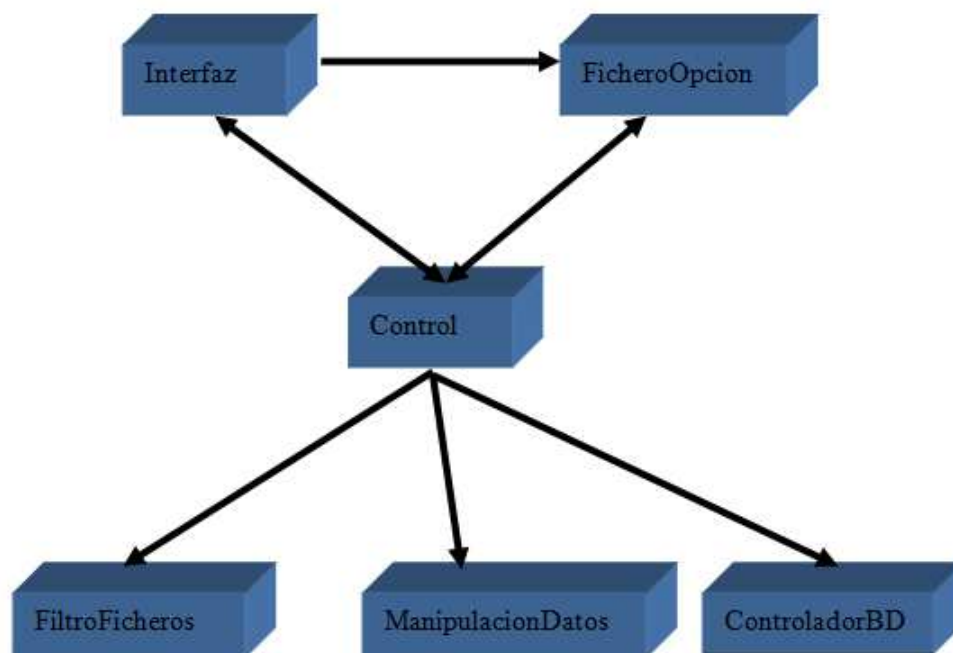


Ilustración 11. Los paquetes de GreedExCol

Por otro lado, se ha utilizado XML y HTML en GreedEx/GreedExCol: XML para la creación de ficheros de datos y HTML para el desarrollo de la ayuda de GreedEx/GreedExCol. El lenguaje de marcado XML (*Extensible Markup Language*) es un estándar abierto para la descripción de los datos de la W3C. HTML es otro lenguaje de marcado para la creación de documentos para las páginas Web. XML utiliza una estructura similar a la de HTML, salvo que en HTML se define cómo se muestran los elementos, mientras que XML define lo que estos elementos contienen.

Para leer los ficheros de datos de entrada o salida en GreedEx se utilizó el API SAX que sirve para acceder a los datos de un documento XML mediante un *Parser* (analizador sintáctico), también se encarga de leer el documento XML de las traducciones necesarias para la internacionalización del sistema. Las visualizaciones se muestran utilizando los API de Java. Por otra parte, para exportar las tablas en formato Excel en GreedEx/GreedExCol, se utilizó la librería JXL que es un código abierto de Java para leer, escribir y modificar hojas de cálculo Excel de forma dinámica. Contiene todos los archivos compilados *. Clase, los metadatos asociados y los recursos que utiliza la API de Java Excel internamente.

Capítulo 4: Evaluaciones de Usabilidad de GreedEx

En este capítulo, presentaremos las evaluaciones de usabilidad de GreedEx que hemos llevado a cabo durante la tesis doctoral. En el primer apartado se hace un breve resumen de las evaluaciones de usabilidad de GreedEx. En el segundo y tercer apartado, se detallan dos evaluaciones de usabilidad de la herramienta y por último, se presenta la evolución de la herramienta GreedEx.

4.1 Evaluación de Usabilidad de GreedEx

Las evaluaciones de usabilidad permiten identificar errores y debilidades. GreedEx se evaluó varias veces (Velázquez-Iturbide, et al., 2013b) (Velázquez-Iturbide, et al., 2013a) con dos métodos diferentes:

1) Las evaluaciones de experto. Varias evaluaciones se llevaron a cabo por el instructor del curso. Se han realizado más de cien pruebas de usabilidad heurística o usabilidad tutorial para mejorar el sistema GreedEx.

2) Dos evaluaciones con los usuarios finales. Estas evaluaciones se llevaron a cabo en las sesiones de laboratorio con los cuestionarios de opinión que nos dio la información sobre la facilidad de uso, eficacia y aceptación del usuario. Conviene aclarar que la calidad reflejada en datos es la calidad “percibida” por los alumnos, no su calidad “objetiva”. Dicha percepción no sólo está condicionada por la aplicación en sí, sino por otros factores, como por ejemplo si se ha explicado adecuadamente la lección, la práctica o la aplicación, o incluso factores del entorno (configuración de los equipos, acceso a Internet, etc.). Además, en las dos evaluaciones de usabilidad, también se observó el comportamiento de los alumnos durante las tres sesiones. La observación fue utilizada como un método de investigación complementaria para obtener información adicional, imposible de obtener con los otros métodos. A continuación, describimos estas dos evaluaciones que hemos llevado a cabo durante este trabajo de tesis, tanto mediante cuestionarios como mediante observación.

4.2 Primera Evaluación de Usabilidad mediante Cuestionarios

Esta evaluación de GreedEx se realizó en noviembre de 2010 (Debdi & Vélazquez-Iturbide, 2011b), en la asignatura troncal “Diseño y Análisis de Algoritmos”, de tercer curso de Ingeniería Informática. Participaron 36 alumnos. El profesor había impartido dos sesiones de teoría. La primera sesión, de 2 horas de duración, incluyó una presentación de las características de los problemas de optimización y de la técnica voraz, así como varios problemas sencillos y una discusión sobre el descubrimiento de funciones de selección óptimas. La segunda sesión fue una clase convencional de una hora sobre el algoritmo de Dijkstra. Los alumnos estaban familiarizados con el método experimental y el asistente GreedEx. La tercera clase fue una sesión de 2 horas. En la primera mitad, el profesor había explicado dos problemas con GreedEx: maximizar el número de objetos en una mochila y maximizar el peso introducido en una mochila. Junto a los problemas, había presentado los conceptos y el proceso de experimentación. Después, los alumnos habían podido experimentar en el aula informática con GreedEx para el resolver óptimamente problema de la mochila.

La evaluación de usabilidad se realizó en una sesión de prácticas posterior, de dos horas de duración. Los alumnos se descargaron del campus virtual todo el material necesario para realizar la práctica: enunciado, modelo de informe y asistente. Al final de la sesión se les entregó en papel un cuestionario de opinión sobre GreedEx con preguntas abiertas y preguntas tipo test, éstas con valores situados en una escala de Likert de 1 (muy mala) a 5 (muy buena). El cuestionario constaba de cuatro tipos de preguntas: Facilidad de uso de GreedEx (2 preguntas, la primera es tipo test y la segunda de tipo preguntas abiertas), Utilidad de GreedEx (2 preguntas tipo test), Calidad de GreedEx (13 preguntas tipo test que incluye la valoración de elementos específicas de GreedEx) y finalmente Preguntas Generales (una pregunta tipo test y 4 preguntas generales). Incluimos el enunciado y el modelo de informe de la evaluación en el Anexo 1, así como el cuestionario de opinión en el Anexo 2.

4.2.1 Valoraciones Numéricas sobre Aspectos Generales

En la

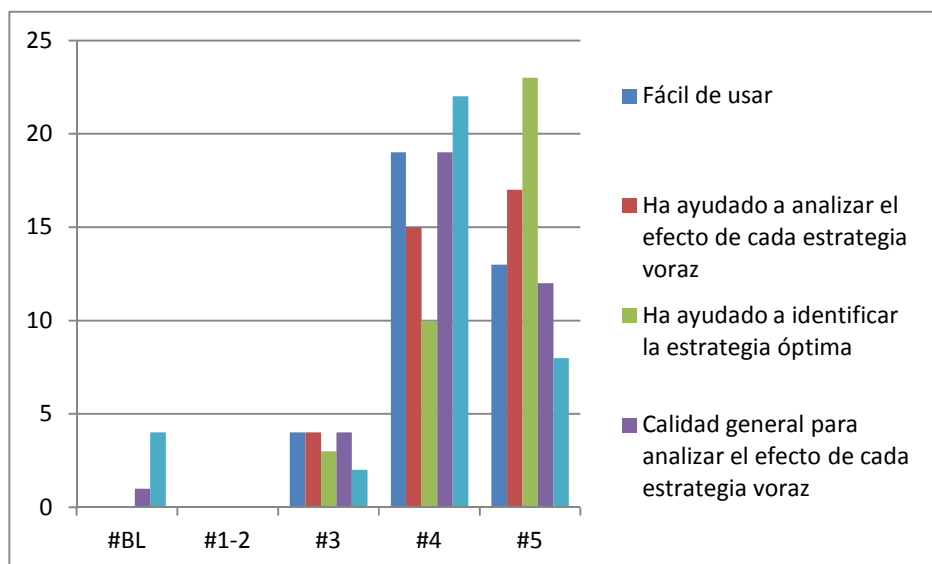
Tabla 17 incluimos los resultados de las preguntas tipo *test* generales.

Tabla 17. Resultados numéricos de las preguntas generales (Primera evaluación)

PREGUNTA	MEDIA	DESV. TÍPICA
Fácil de usar	4.29	0.60
Ha ayudado a analizar el efecto de cada estrategia voraz	4.43	0.69
Ha ayudado a identificar la estrategia óptima	4.57	0.69
Calidad general para analizar el efecto de cada estrategia voraz	4.26	0.66
En conjunto te ha gustado GreedEx	4.12	0.52
Media total	4.34	

Destacan las respuestas obtenidas en la tercera pregunta (utilidad para la identificación de la estrategia óptima) con una media de 4.57. El resultado más bajo se ha obtenido en la última pregunta (en conjunto te ha gustado GreedEx) con una media de 4.12.

Se agruparon las respuestas de las cinco preguntas por su puntuación recibida. Tal y como se puede ver en la Ilustración 12, los valores obtenidos han sido bastante altos. Las 179 respuestas se desglosan 157 altas (puntuación de 4) o muy altas (puntuación de 5), 17 regulares (puntuación de 3) y 5 en blanco. Además, se observa que la moda de las preguntas de utilidad de GreedEx (ha ayudado a analizar el efecto de cada estrategia voraz y ha ayudado a identificar la estrategia óptima) es de 5 mientras que el resto de preguntas tienen una moda de 4.

**Ilustración 12.** Valoración sobre aspectos generales (Primera evaluación)

4.2.2 Valoraciones numéricas sobre Aspectos Concretos

La Tabla 18 muestra la evaluación de calidad de elementos concretos de GreedEx, ordenados de mayor a menor.

Tabla 18. Resultados ordenados de la calidad de sus elementos (Primera evaluación)

PREGUNTA	#BL	#1-2	#3	#4-5	MEDI A	DESV. TÍPICA
Pestaña de tabla de resumen global	0	0	6	22	4.14	0.76
Pestaña de tabla de datos de entrada	0	0	6	22	4.11	0.74
Introducción o generación de datos de entrada	0	2	6	20	4.07	0.98
Estructura del menú principal	0	0	4	24	4.04	0.58
Pestaña de tabla de resultados	0	0	7	21	4.00	0.72
Ejecución/Animación del algoritmo	0	0	7	21	4.00	0.72
Selección de estrategias	0	14	7	20	4.00	0.86
Pestaña de algoritmo	0	3	5	20	3.93	0.98
Pestaña de problema	0	2	7	19	3.89	0.92
Panel de visualización	0	1	8	19	3.89	0.83
Iconos	0	2	11	15	3.61	0.83
Facilidades de exportación	0	25	5	9	2.82	1.93
Total de respuestas	0	49	128	232	3.92	0.90

Los resultados sobre la calidad de elementos concretos son buenos. Como puede verse en la Tabla 14, el resultado medio de la evaluación es 3.92. De un total de 409 respuestas, 49 malas o muy malas (11.98%), 128 regulares (31.29%) y 232 buenas o muy buenas (56.72%).

Se debe indicar que las diferencias grandes entre las medias de los elementos concretos (especialmente facilidades de exportación con el resto de elementos, desde 2.82 a 4.14) se debe a que la exportación de tablas no fue soportado por GreedEx (el aspecto más solicitado con diferencia) dado que los alumnos necesitan exportar las tablas para incluirlos en el informe de la práctica. Cabe mencionar que en la valoración de las facilidades de exportación en el cuestionario se refería a la exportación de imagen, secuencia de imágenes, animación y estrategias que fueron soportados por GreedEx en el momento de la evaluación. Los elementos mejor valores han sido las tabla de resumen y datos de entrada. Los peores fueron las facilidades de exportación.

4.2.3 Críticas y Sugerencias Recibidas

Realizamos cuatro preguntas de respuesta abierta para tener una opinión detallada de los alumnos sobre los aspectos positivos y negativos de GreedEx. La Ilustración 13 contiene el número de respuestas en blanco y escritas para cada pregunta abierta. Según esta Ilustración 8, podemos extraer las siguientes conclusiones:

- Preguntas sobre las que la mayoría de los alumnos no tienen comentarios que realizar: partes poco útiles que suprimir.
- Preguntas similares en respuestas escritas y respuestas en blanco: aspectos negativos y partes difíciles de usar.
- Preguntas sobre las que la mayoría de los alumnos tienen comentarios que realizar: partes útiles de las que carece GreedEx y aspectos positivos.

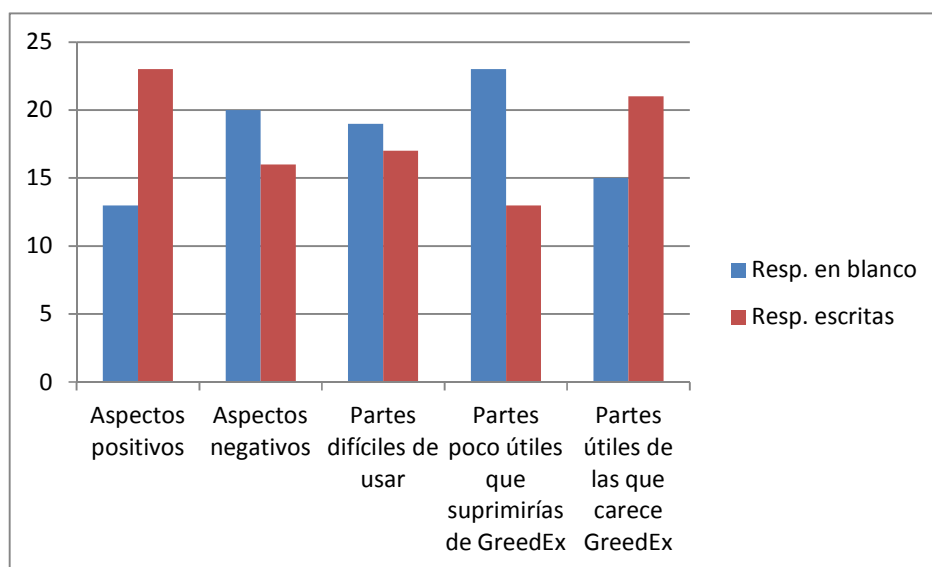


Ilustración 13. Preguntas de respuesta abierta (Primera evaluación)

La pregunta que indagaba sobre los aspectos positivos de GreedEx obtuvo un total de 23 respuestas. Hemos agrupado y ordenado las respuestas de mayor a menor, dando el número de respuestas para aquellos aspectos con más de una cita:

- Facilidad de uso (7 respuestas),

- Análisis y la comprensión de los efectos de las distintas funciones de selección para entender bien el algoritmo (6 respuestas)
- Las distintas funciones de ejecución (5 respuestas)
- El apoyo de GreedEx en resolver el problema planteado (2 respuestas)
- Interfaz de usuario (2 respuestas)
- Visualización (2 respuestas)
- Animación del algoritmo (2 respuestas)
- Generación aleatoria de los datos de entrada (3 respuestas)
- Tablas (2 respuestas)

Cuando se preguntó a los alumnos sobre los aspectos negativos de GreedEx hubo 15 respuestas, la mayoría de ellos señalaron la ausencia de la funcionalidad de la exportación de tablas en la herramienta (7 respuestas) dado la necesidad de exportar los resultados obtenidos al informe final de la práctica. También hubo alumnos que indicaron algunos errores de la aplicación (4 respuestas) y el funcionamiento incorrecto de GreedEx en el sistema MAC OS.

En la pregunta sobre la identificación de partes más difíciles de usar en GreedEx hubo 17 respuestas, de las que 6 señalan que es muy sencillo o “nada”. Hubo alumnos que destacaron la función de exportación de capturas (3 respuestas). Otros consideraron como poco significativos e intuitivos los iconos del sistema (2 respuestas). Quedaron por detrás otras cuestiones como el formato de visualización (2 respuestas) o la ejecución guiada, los atajos del teclado y la inserción de datos.

Respecto a la pregunta que pedía identificar partes de GreedEx que el alumno suprimiría, hubo 23 respuestas en blanco, 6 respuestas que afirman no suprimirían nada. Se destacó la pestaña de la tabla abreviada por la poca información que contiene (3 respuestas) y la exportación de la secuencia de imágenes. Otros alumnos indicaron la inutilidad de la pestaña de descripción del problema (que podría ir como comentario en el algoritmo según comenta algún alumno).

Como complemento, se preguntaba por otras funciones que sería útil incorporar a GreedEx. Se recogieron 21 respuestas, se destacó por excelencia la exportación de tablas (15 respuestas) debido a su importancia para el desarrollo del informe final de la práctica. Algunos alumnos señalaron la visualización de los paneles de GreedEx que podría mejorarse, incluyendo la posibilidad de mover o minimizar los tres paneles principales de GreedEx (3 respuestas), sobre todo maximizar la tabla de

resumen. Otras peticiones fueron la incorporación de un pequeño GreedEx al IDE Eclipse Gavab de la URJC y la eliminación de la ejecución guiada.

4.2.4 Conclusiones de la Primera Evaluación de GreedEx

Podemos resumir los resultados de la evaluación de GreedEx en las siguientes conclusiones generales

- Lo más valorado de GreedEx es su facilidad de uso y su ayuda para comprender, analizar o comparar las distintas estrategias. También destacan su utilidad en general y sus funciones de ejecución.
- La mayor parte de los alumnos identifican aspectos positivos de GreedEx, pero también partes útiles de las que carece. El aspecto más solicitado con diferencia fue mejorar la exportación.
- Los elementos mejor valorados de GreedEx son las pestañas de tabla con resumen global y de datos de entrada, la introducción o generación de datos de entrada y la estructura de la interfaz de usuario. Sin embargo, la diferencia de valoración con el resto de elementos es escasa. La excepción la constituyen los iconos y sobre todo las facilidades de exportación.

En resumen, los resultados han sido positivos tanto por la valoración de los alumnos como por la información recogida para mejorar GreedEx. Esta evaluación sirvió para revisar algunos aspectos para soportar una buena parte de las sugerencias recogidas (exportación de tablas) y corregir los problemas notificados.

4.3 Primer Análisis de las Observaciones

Para obtener más información sobre las dificultades que pueden experimentar los alumnos, se registraron observaciones en tres sesiones de laboratorio durante la primera evaluación de la usabilidad (detallada en el apartado anterior). Se observó el comportamiento de los alumnos como un método de investigación complementario para obtener información adicional, imposible de obtener con los otros métodos (Debdi, et al., 2012).

Recordemos que la primera sesión sirvió para que los alumnos se familiaricen con el método experimental y el sistema GreedEx y encontrar las funciones de selección óptimas para el problema de la mochila. En la segunda sesión, los alumnos tenían que resolver la misma tarea para el problema de la selección de actividades. Al final de la segunda sesión, los alumnos tenían que entregar un

informe de sus resultados, así como rellenar un cuestionario de opinión sobre GreedEx. Finalmente, en la tercera sesión, se identificaron las funciones de selección óptimas para el problema de selección de actividades, en este caso los alumnos tuvieron que explicar los errores que hicieron en la segunda sesión. El número de participantes en cada sesión fue de 22, 36 y 18, respectivamente.

Podemos diferenciar dos tipos de observaciones: preguntas de los alumnos al profesor y la actividad de los alumnos mientras usaban la herramienta. Cabe mencionar que los dos tipos de observaciones no sólo tienen diferentes objetivos, sino también características diferentes. Por un lado, cuando un alumno hace una pregunta al profesor significa que está haciendo explícitas sus dificultades. Por otra parte, las observaciones recogidas mientras un alumno interactúa con el ordenador proporcionan datos más oscuros, lo cual podemos deducir la intención del alumno mediante sus actos. Hubo tres observadores en total:

- Un observador que registró las preguntas que los alumnos hicieron al profesor. También registró los eventos o conductas que podrían ser interesantes, por ejemplo, el uso de una aplicación de software diferente.
- Dos observadores tomaron nota de la actividad de los alumnos. Cada observador fue asignando de 2 a 4 grupos. Para cada grupo, una observación se registró cada cinco minutos.

4.3.1 Resumen de las Observaciones

Reunimos 55 observaciones, que corresponden a 30 alumnos. El número de observaciones a partir de la primera, segunda y tercera sesión fueron 16, 35 y 4, respectivamente.

1) Preguntas de los Alumnos

Las preguntas hechas por los alumnos fueron analizadas y clasificadas en varias categorías. Podemos clasificar las observaciones en cuatro categorías:

- Observaciones sobre el asistente interactivo GreedEx, por ejemplo, acerca de los elementos específicos de GreedEx (19 preguntas).
- Observaciones sobre otros usos, por ejemplo, navegar por Internet (4 observaciones).

- Observaciones sobre el problema que hay que resolver, por ejemplo, dudas sobre el enunciado del problema (22 preguntas).
- Observaciones sobre otros acontecimientos potencialmente relevantes, por ejemplo, alumnos tomando notas en papel (10 observaciones).

Se analiza el contenido de las observaciones correspondientes en cada categoría. Con respecto a las observaciones sobre GreedEx, distinguimos varios tipos:

- 3 observaciones relacionadas con la instalación o problemas con la ejecución de GreedEx, por ejemplo, el sistema se bloquea.
- 5 preguntas dirigidas a la comprensión de los elementos específicos de GreedEx, por ejemplo, partes de la visualización. Un alumno, incluso puso en duda la fiabilidad de los resultados de GreedEx.
- 3 preguntas sobre la forma de proceder, por ejemplo, en un cuadro de diálogo.
- 4 preguntas acerca de la dinámica de GreedEx, por ejemplo, ¿por qué se desactivarán algunas opciones.

Por otra parte, hay 4 observaciones de los alumnos que utilizan otras herramientas de software, como el IDE Eclipse y Google.

Con respecto a las 22 observaciones sobre el problema a resolver, hay seis tipos:

- 7 preguntas sobre el enunciado del problema.
- 5 preguntas sobre la relación entre optimalidad y los conjuntos de datos.
- 3 preguntas sobre los datos de entrada que se utilizan para la experimentación.
- 2 alumnos que tenían dificultades para encontrar contraejemplos para una función de selección no óptima.
- 4 alumnos que querían saber si estaban bien dirigidos.
- 1 observación de una pareja de alumnos que experimentaron con otro problema de GreedEx.

Por último, se han realizado 10 observaciones de otros eventos:

- 2 observaciones de la conducta de los alumnos, por ejemplo, hablando entre sí.

- 8 observaciones de alumnos que toman notas.

El análisis de las preguntas de los alumnos muestra que algunas preguntas son habituales en una situación de laboratorio y no requieren ninguna intervención, por ejemplo, alumnos que quieren saber si fueron bien dirigidos. Otras preguntas refuerzan algunos de los resultados obtenidos con otros métodos de evaluación y otras proporcionan información adicional que puede ser utilizado para mejorar el proceso de enseñanza. Además, hemos tenido algunos resultados inesperados:

- La necesidad de revisar (y quizás arreglar o mejorar) algunos elementos específicos de GreedEx, principalmente algunos cuadros de diálogo.
- Un malentendido potencial de los datos que aparecen en la visualización del problema de la mochila.
- El uso de notas para registrar los datos.
- La búsqueda de soluciones en la Web. Este comportamiento parece estar destinado a hacer trampa pero no obtuvo evidencia de esto cuando se analizó el informe presentado por estos alumnos.

Por último, algunas observaciones podrían ser utilizadas para otros fines. En particular, algunas observaciones sobre el comportamiento de los alumnos pueden ser utilizados para distinguir notas en parejas de alumnos. Por ejemplo, un alumno con un comportamiento pasivo, mientras que su compañero de prácticas está más involucrado.

2) Observación de la Actividad

Reunimos 35 observaciones de 21 alumnos organizados en 14 grupos (ya sea individual o parejas). El número de alumnos que se observaron en cada sesión fue de 12, 11 y 12 respectivamente. El proceso de la recogida de las observaciones de la actividad de los alumnos es el siguiente:

- En la primera etapa, se elaboró un mapa con los datos sobre la asistencia del alumno a las tres sesiones. El mapa de cada sesión nos permitió conocer la asistencia y la proximidad de alumnos. La posición de cada alumno se llenó con su nombre y varios números obtenidos de varias fuentes: número asignado por orden alfabético, número de inscripción

en el curso, el número asignado a su informe y el número asignado a su cuestionario (si se conoce).

- En la segunda etapa, se crearon registros de observaciones sobre cada grupo de alumnos. Usando los números disponibles en los mapas de sesión, se recogieron información de diferentes fuentes (si está disponible): los comentarios del profesor sobre la práctica, resultados y respuestas dadas por el alumno en el cuestionario, observaciones registradas en las diferentes sesiones y las preguntas al profesor.
- En la tercera etapa, se limpió el registro de observación de cada grupo de alumnos. Se eliminaron los números de identificación y otra información irrelevante. Además, se han suprimido las observaciones de otros alumnos que no tenían relación con el grupo.

Las observaciones de actividad de los alumnos recogidas fueron analizadas relacionándolos con datos obtenidos con otros métodos, es decir, el rendimiento del grupo (de acuerdo con el análisis de su informe), la satisfacción con GreedEx (de acuerdo con los cuestionarios de usabilidad) e incluso preguntas al profesor. Este análisis se realizó en varias etapas. Por último, los datos asociados a cada grupo se analizaron y se compararon. Algunas de las acciones que facilitaron el análisis fueron:

- Resumir las observaciones en períodos de tiempo.
- Relacionar algunas observaciones con las preguntas que hicieron los alumnos al profesor, sus respuestas al cuestionario o el contenido de sus informes.

La mayoría de las observaciones de la actividad de los alumnos fueron consistentes con sus calificaciones de la práctica, con sus evaluaciones subjetiva de la usabilidad de GreedEx y con sus preguntas al profesor. Por lo tanto, hemos ganado más evidencia en estos resultados. Por ejemplo:

- Los alumnos que pasan mucho tiempo para capturar y manipular imágenes o tablas en GreedEx para el informe de la práctica han valorado negativamente la facilidad de exportación de imágenes/tablas de GreedEx en el cuestionario.
- Los alumnos con calificaciones bajas o altas presentan un comportamiento esperado, por ejemplo, que dedican menos tiempo a la experimentación que la redacción del informe.

Las observaciones fueron más claras con respecto a la utilización de otras herramientas diferente de GreedEx. Las observaciones más relevantes eran:

- Cuatro grupos tomaron notas. Por lo general los alumnos toman nota de los resultados de ejecución, pero también se observó un alumno dibujando diagramas.
- Ocho grupos capturaron fotos de la pantalla. En seis de estos grupos, se observó la manipulación de las capturas con Paint. Además, un grupo utilizó Excel para crear sus cuadros que más tarde fueron capturados y manipulados.

4.3.2 Conclusiones de las Observaciones

Se han obtenido algunos nuevos resultados interesantes relacionados con el método didáctico:

- Dudas y malentendidos de los alumnos sobre el planteamiento del problema, el método experimental, algunos elementos de GreedEx o el proceso de trabajo con GreedEx.
- La necesidad de revisar (y quizás arreglar o mejorar) algunos elementos específicos de GreedEx.
- La necesidad de explicar mejor algunos componentes de GreedEx.

Finalmente, se obtuvieron algunos resultados que son útiles para la evaluación o para comprender mejor la dinámica de los alumnos. En algunos casos, merecen mayor atención en las futuras sesiones:

- El uso de notas, sobre todo para grabar datos, pero probablemente con otros propósitos también.
- Navegación Web, posiblemente en busca de soluciones de la tarea.
- Algunas observaciones sobre el comportamiento de un grupo de alumnos pueden ser utilizados para calificarles de manera diferente. Por ejemplo, un alumno que se observó con un comportamiento pasivo, mientras que su compañero estaba haciendo la práctica.

También se ha obtenido información útil sobre cómo llevar a cabo con eficacia las observaciones en el futuro para mejorar las herramientas interactivas. Se registraron dos tipos de

observaciones: preguntas al profesor y la actividad de los alumnos. Sin embargo, encontramos mucho más útil la grabación de preguntas. Nos dieron un vívido retrato de las dificultades específicas de los alumnos. Con respecto a las observaciones de la actividad de los alumnos, también podemos diferenciar dos tipos de observaciones. Las observaciones más útiles fueron relacionadas con acciones diferentes del uso de GreedEx, por ejemplo tomar notas o el uso de Paint. Sin embargo, el registro de la actividad con GreedEx no era lo suficientemente detallada para que tenga sentido. Además, el esfuerzo humano necesario para grabar cualquier tipo de observación era enorme. Aunque hay varias alternativas de grabación (por ejemplo, mediante el uso de cámaras de vídeo), el mejor enfoque parece ser el registro del equipo y el análisis de las interacciones del usuario con la herramienta. En resumen, se propone utilizar el registro para registrar las interacciones con el sistema y observaciones para registrar otras actividades (preguntas, comportamiento social o el uso de otros sistemas).

Por otra parte, las observaciones recogidas han sido de gran ayuda para mejorar los diferentes elementos de nuestro enfoque: la introducción de cambios en el contenido y el énfasis de las clases, la introducción de cambios en la organización de las sesiones de laboratorio, la elaboración de nuevas transparencias y la mejora de algunos elementos del asistente interactivo GreedEx.

4.4 Segunda Evaluación de Usabilidad mediante Cuestionarios

Esta evaluación de GreedEx se realizó en octubre de 2011 (Debdi & Velázquez, 2012b), en la asignatura troncal “Diseño y Análisis de Algoritmos” de tercer curso de Ingeniería Informática. Participaron 10 alumnos. El procedimiento, la práctica y el cuestionario de usabilidad son los mismos que la evaluación anterior. Incluimos el enunciado y el modelo del informe de la evaluación en el Anexo 1, así como el cuestionario de opinión en el Anexo 2.

4.4.1 Valoraciones Numéricas sobre Aspectos Generales

En la Tabla 19 incluimos los resultados de las preguntas tipo *test* generales.

Tabla 19. Resultados numéricos de las preguntas generales (Segunda evaluación)

PREGUNTA	MEDIA	DESV. TÍPICA
Fácil de usar	4.50	0.53

Ha ayudado a analizar el efecto de cada estrategia voraz	4.63	0.74
Ha ayudado a identificar la estrategia óptima	4.63	0.74
Calidad general para analizar el efecto de cada estrategia voraz	4.00	0.93
En conjunto te ha gustado GreedEx	4.25	0.46
Media total	4.40	

Destacan las respuestas obtenidas en las preguntas primera a tercera (facilidad de uso, utilidad para analizar el efecto de cada estrategia y utilidad para identificar la estrategia óptima), con medias de 4.5, 4.63 y 4.63. El resultado más bajo se ha obtenido en la cuarta pregunta (calidad general para analizar el efecto de cada estrategia voraz). Se deduce que los alumnos han percibido a GreedEx más útil que bueno para la tarea recomendada.

Como puede observarse en la Ilustración 14, los valores obtenidos han sido bastante altos. Las 40 respuestas se desglosan 37 altas (valoración de 4) o muy altas (valoración de 5), 2 regulares (valoración de 3) y 1 mala o muy mala (valoración de 2 o 1). Además, se puede ver que hay un sesgo aun más claro que en la evaluación anterior entre preguntas cuya moda es 4 (en conjunto te ha gustado GreedEx y calidad general para analizar el efecto de cada estrategia voraz) y las que tienen una moda de 5 (ha ayudado a analizar el efecto de cada estrategia voraz y ha ayudado a identificar la estrategia óptima).

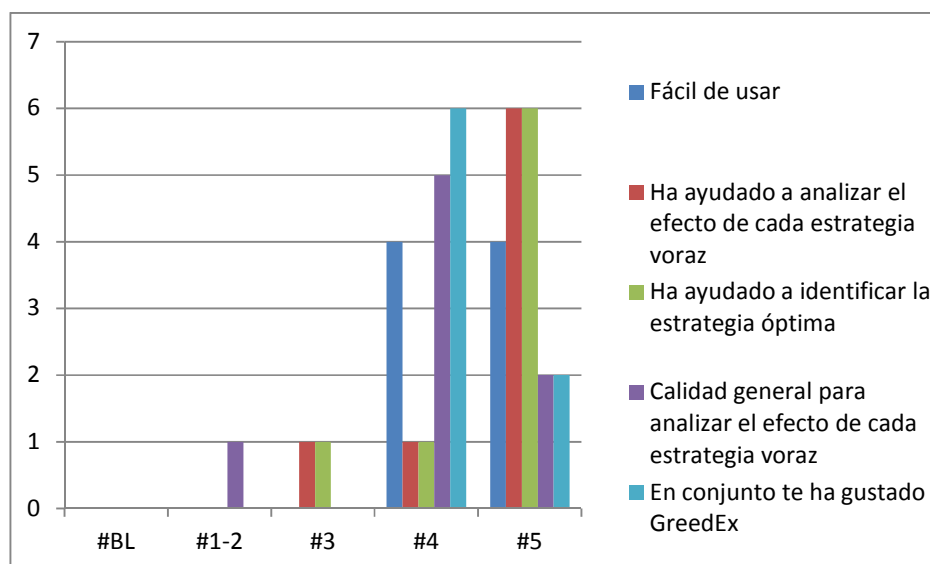


Ilustración 14. Valoración sobre aspectos Generales (Segunda evaluación)

4.4.2 Valoraciones Numéricas sobre Aspectos Concretos

La Tabla 20 muestra la evaluación de calidad de elementos concretos de GreedEx, ordenados de mayor a menor.

Tabla 20. Resultados numéricos de los aspectos concretos (Segunda evaluación)

PREGUNTA	MEDIA	DESV. TÍPICA
Estructura del menú principal	4.38	0.52
Funciones de selección de estrategias	4.38	0.74
Pestaña de tabla de abreviada	4.25	0.71
Iconos	4.25	0.71
Panel de visualización	4.25	0.89
Pestaña de tabla de resumen	4.13	0.64
Pestaña de problema	4.13	0.64
Pestaña de algoritmo	4.00	0.76
Ejecución/Animación del algoritmo	4.00	0.82
Pestaña de tabla de resultados	3.88	0.64
Introducción o generación de datos de entrada	3.63	0.92
Exportación de imágenes y tablas	3.57	1.27
Pestaña de tabla de datos de entrada	3.50	1.20
Total de respuestas	4.01	

Los resultados sobre la calidad de elementos concretos son buenos. Como puede verse en la Tabla 20, el resultado medio de la evaluación es 4.01. De un total de 96 respuestas, 4 son malas o muy malas (4.17%, con una valoración de 2 o 1), 17 regulares (17.71%, con una valoración de 3), 73 buenas o muy buenas (76.04%, con una valoración de 4 o 5) y 2 en blanco (2.08%). Se observa una mejoría en la valoración de la mayor parte de los elementos, salvo las pestañas de las tablas de resultados, de datos de entrada y del problema y la introducción o generación de datos de entrada dado las mejoras realizadas tras la primera evaluación.

4.4.3 Críticas y Sugerencias Recibidas

La mayor parte de los alumnos identificaron aspectos positivos y negativos de GreedEx, pero pocos comentaron sobre elementos difíciles de usar, suprimibles o de los que carezca GreedEx. La Ilustración 16 contiene el número de respuestas en blanco y escritas para cada pregunta abierta.

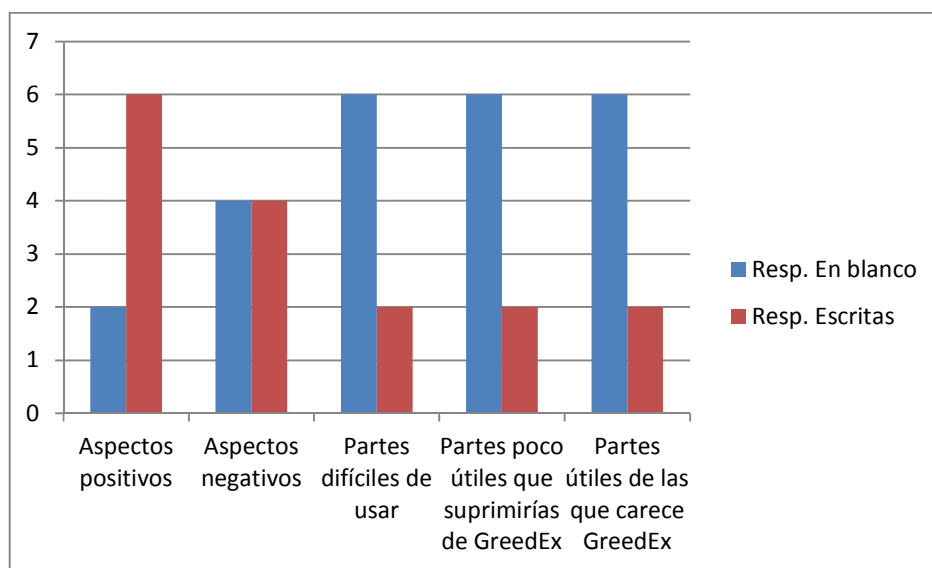


Ilustración 15. Preguntas de respuesta abierta (Segunda evaluación)

Una pregunta indagaba sobre los aspectos positivos de GreedEx, habiendo obtenido un total de 6 respuestas concretas. La facilidad de uso (2 respuestas) y la función de exportación (2 respuestas) fueron las dos cualidades destacadas que mencionaron los alumnos en esta pregunta. También hubo quien señaló la facilidad de detectar las funciones de selección óptimas y el análisis de las mismas para la comprensión del algoritmo.

Con respecto a los aspectos negativos, hay un total de 4 preguntas: los alumnos volvieron a mencionar la interfaz de usuario pidiendo agrandar la visualización (2 respuestas). Se recogió 2 respuestas sobre errores en la exportación de tablas y el almacenamiento de los logs. También hubo quien mencionó la ausencia de las teclas de atajo para las funciones de ejecución y la generación restringida de datos de entrada.

En la pregunta sobre la identificación de las partes más difíciles de usar hubo 2 respuestas, una que alagaba al sistema y sólo una que identificaba la función de la ejecución intensiva. De las partes que el alumno suprimiría hubo dos respuestas pero de nuevo sólo una identifica algo concreto, en este caso la ejecución paso a paso. Finalmente, en las principales sugerencias sobre los partes que incorporar en GreedEx, se recogieron 2 respuestas que fueron las teclas de atajo para las funciones de ejecución y la exportación de imágenes.

Se puede observar que en los resultados de esta evaluación, la exportación de tablas se recogió como un aspecto positivo, mientras que en la anterior evaluación se recogió como aspecto que debe incorporarse en el sistema GreedEx. También se modificó la ejecución guiada que se recogió como sugerencia en la evaluación anterior, esta opción fue modificada añadiendo la opción de realizar una ejecución libre en esta última evaluación.

4.4.4 Conclusiones de la Segunda Evaluación de GreedEx

Los resultados han sido positivos, tanto la valoración de los alumnos como la información recogida para mejorar GreedEx. Podemos resumir los resultados de esta evaluación de usabilidad en las siguientes conclusiones generales:

- Los aspectos mejor valorados son la interfaz de usuario (menús e iconos), selección de estrategias, tabla abreviada y visualización. Los peor valorados son las tablas de datos de entrada y de resultados, exportación y generación de datos de entrada.
- La mayor parte de los alumnos identifican aspectos positivos y negativos de GreedEx, pero pocos elementos difíciles de usar, suprimibles o de los que carezca GreedEx.

Por otra parte, algunas dificultades detectadas en esta evaluación de usabilidad convencieron de la necesidad de revisar algunos elementos en GreedEx como:

- Revisión de la interfaz de usuario (legibilidad y teclas de atajo).
- Revisión de las funciones de exportación de tablas.
- Revisión de la creación de ficheros de logs.

4.5 Segundo Análisis de las Observaciones

La organización y el procedimiento de este análisis fueron igual que las observaciones anteriores. Estas observaciones han sido recogidas durante las sesiones del laboratorio de la segunda evaluación de usabilidad descrita en el apartado anterior.

Cabe mencionar que han sido usados los mismos métodos de observación de la primera evaluación salvo que en esta evaluación se han incluido los logs como una fuente de información más.

4.5.1 Resumen de las Observaciones

En esta evaluación han participado 10 alumnos y se crearon 7 grupos, aunque han sido observados solamente los 4 primeros grupos. Las conclusiones han sido obtenidas después de analizar diferentes fuentes de información (logs, cuestionarios y observaciones).

1) Preguntas de los Alumnos

En total hubo 23 preguntas recogidas. Las preguntas hechas por los alumnos fueron analizadas y clasificadas en varias categorías. Podemos clasificar las observaciones en cuatro categorías:

1. Preguntas sobre el asistente interactivo GreedEx, por ejemplo, acerca de los elementos específicos de GreedEx (6 preguntas).
2. Preguntas sobre otros usos de la computadora, por ejemplo, navegar por Internet (6 preguntas).
3. Preguntas sobre el problema que hay que resolver, por ejemplo, dudas sobre el enunciado del problema (7 preguntas).
4. Preguntas sobre cómo encontrar un contraejemplo para una función de selección no óptima (4 preguntas).

Algunos puntos interesantes que han sido recogidas por los alumnos fueron sus preguntas sobre elementos muy sencillas o obvias en nuestro parecer, por ejemplo que algunos alumnos no entendían el significado de la cardinalidad o el significado de los diferentes valores del dialogo del tiempo en el problema de selección de actividades.

2) Observación de la Actividad

En resumen, algunas observaciones no eran útiles, ya sea porque son demasiado escasos o que muestran malentendidos normales de los alumnos. Sin embargo, otras observaciones fueron fructíferas para obtener más información sobre nuestro método experimental, el sistema GreedEx y el curso en sí. Las observaciones más relevantes eran:

- No usar la opción de la ejecución intensiva que pudiera ayudar a descartar la tercera función de selección como una función de selección óptima.
- El uso de papel y lápiz para entender el algoritmo.

Otras observaciones que han sido fruto de las diferentes fuentes de información recogidas (el informe entregado, el cuestionario, las observaciones recogidas y los logs) son:

- Se observó una contradicción entre el cuestionario de usabilidad y los logs en un grupo, donde se evaluó positivamente la facilidad de exportación sin usar esta funcionalidad según los logs recogidos.
- Otro grupo evaluó positivamente la funcionalidad de la exportación de tablas, pero en su informe ha incluido imágenes usando la herramienta Paint para exportar las tablas.

4.5.2 Conclusiones de las Observaciones

Los siguientes puntos son algunas conclusiones o sugerencias que podíamos incluir en las futuras sesiones:

1. La necesidad de explicar mejor algunos componentes de GreedEx:
 - Añadir en la primera sesión de la introducción de GreedEx la explicación de la ejecución intensiva y su utilidad a la hora de deducir las funciones de selección óptimas.
 - Incluir una explicación de la cardinalidad.
 - Añadir la explicación del significado del tiempo total de las actividades.
2. Se obtuvieron algunos resultados que son útiles para la mejora del sistema GreedEx:
 - Mejorar la generación de los datos aleatorios, limitando la pregunta del alumno solamente por el número total de actividades deseadas, eliminando la pregunta por el tiempo máximo, dejando así que el sistema lo determine.
 - Como la mayoría de los alumnos tienen la tendencia a introducir por teclado los datos del ejemplo de la práctica, sería conveniente añadir en el Zip de GreedEx un fichero

de datos de este ejemplo (y que el profesor lo comenta antes de empezar la sesión de que se trata de un ejemplo y que no hay que resolverlo), así el alumno podrá cargarlo y ejecutarlo sin perder tiempo y sobre todo no pensar de que el objetivo de la práctica es resolver solamente este caso.

- Sería conveniente añadir un JDialog, informando sobre el significado de la coloración de los objetos de oscuro a claro porque la mayoría de los alumnos no se dan cuenta que este último es el orden de la elección de los objetos.

Estas mejoras se pueden implementar por medio de la intervención en la dinámica del aula o la ampliación de los materiales escritos (Apuntes) y la mejora del sistema GreedEx. Por otra parte, los hallazgos de las observaciones recogidas fueron consistentes con los resultados obtenidos por el método de evaluación de usabilidad que refuerza la evidencia de los resultados anteriores sobre las debilidades del método experimental o el sistema GreedEx.

4.6 Evolución de GreedEx

Las respuestas sobre aspectos generales y concretos y finalmente realizamos un análisis detallado de la evolución de GreedEx (Velázquez-Iturbide, et al., 2013a)(Velázquez-Iturbide, et al., 2012).

4.6.1 Evolución de Aspectos Generales de GreedEx

La Tabla 21 muestra la evolución de las respuestas sobre aspectos generales de GreedEx en las dos evaluaciones.

Tabla 21. Evolución de los resultados numéricos de las preguntas generales

PREGUNTA	MEDIA	MEDIA	MODA	MODA	MEDIA
	1	2	1	2	TOTAL
Ha ayudado a identificar la estrategia óptima	4.57	4.63	5	5	4.60
Ha ayudado a analizar el efecto de cada estrategia voraz	4.43	4.63	5	5	4.53
Fácil de usar	4.29	4.50	4	4,5	4.39
Calidad general para analizar el efecto de cada estrategia voraz	4.26	4.00	4	4	4.13
En conjunto te ha gustado	4.12	4.25	4	4	4.18

Puede observarse tanto en la Tabla 21 como la Ilustración 16 que ha habido una mejora de todos los criterios con respecto a la evaluación anterior (utilidad, facilidad de uso y satisfacción) salvo la calidad general para analizar el efecto de cada estrategia voraz.

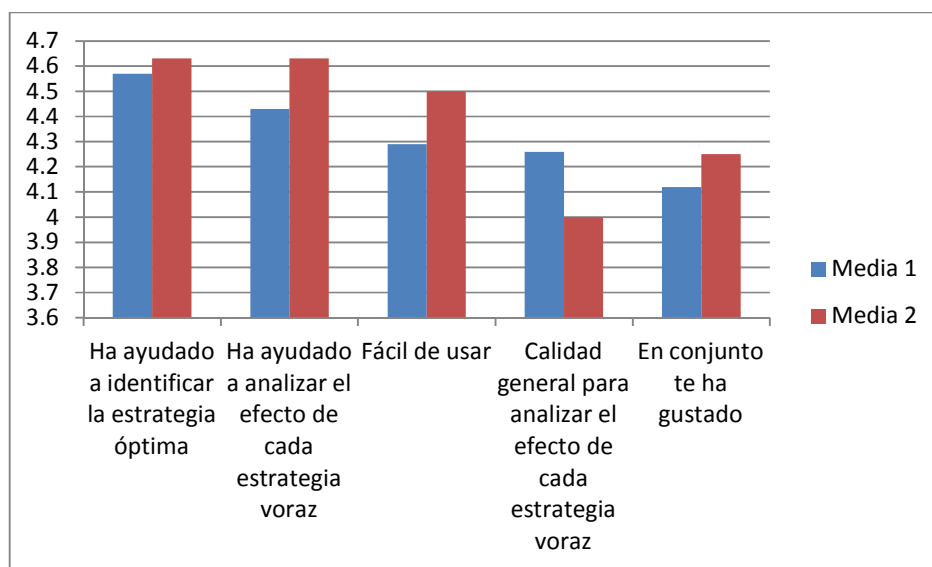


Ilustración 16. Evolución de los aspectos generales de GreedEx

En términos globales, los criterios más valorados son la utilidad para la evaluación y la utilidad para el análisis. En términos medios quedan la calidad general y la facilidad de uso. El criterio peor valorado es la satisfacción global, aunque con un valoración de 4.18.

4.6.2 Evolución de Aspectos Concretos de GreedEx

La Tabla 22 muestra la evolución de los resultados obtenidos para estos elementos a través de las dos evaluaciones.

Tabla 22. Evolución de los aspectos concretos de GreedEx

PREGUNTA	MEDIA EVAL. 1	MEDIA EVAL. 2	MEDIA GLOBAL
Estructura del menú principal	4.04	4.38	4.21
Selección de estrategias	4.00	4.38	4.19
Pestaña de tabla abreviada	4.14	4.25	4.19
Pestaña de tabla con resumen global	4.14	4.25	4.19
Panel de visualización de datos	3.89	4.25	4.07

Pestaña de problema (explicación)	3.89	4.13	4.01
Iconos	3.61	4.25	3.93
Facilidades de exportación / Formato de almacenamiento en fichero de resultados	2.82	3.57	3.19
Ejecución/animación del algoritmo	4.00	4.00	4
Pestaña de algoritmo (código)	4.00	4.00	4
Pestaña de tabla con resultados	4.00	3.88	3.94
Introducción o generación de datos de entrada	4.07	3.63	3.85
Pestaña de tabla con datos de entrada	4.11	3.50	3.80

Como se puede observarse en la Tabla 18, la mayor parte de los elementos de GreedEx han mejorado (8 elementos), otros dos elementos han mantenido igual (ejecución y pestaña de algoritmo). Finalmente tres elementos hayan descendido su valoración (pestaña de tabla de resultados, introducción o generación de datos de entrada y pestaña de tabla con datos de entrada). No encontramos explicación ninguna sobre el porqué de su deterioro. En términos globales, hay seis elementos con una valoración superior a 4, seguidos de cuatro con aproximadamente 4 y tres con una valoración superior que 3. Sin embargo, hay que destacar que los elementos menos valorados son menos importantes para el método experimental, más estáticos o poco usados. La excepción sería la exportación que, objetivamente, ha estado mal soportada.

4.6.3 Análisis de la Evolución de GreedEx

Según la Tabla 21 y Tabla 22, algunas medias de las características y preguntas analizadas sobre GreedEx son mejor, peor o igual en ambas evaluaciones. Para verificar que estas medias son poblacionalmente significativas hemos realizado una comparación de los resultados de las dos evaluaciones sobre las medias poblacionales de estas preguntas y características, realizando los siguientes pasos sobre las dos evaluaciones:

1. Verificación de la normalidad de las muestras.
2. Verificación de la homogeneidad de las muestras o la prueba de Kruskal Wallis en el caso de la no normalidad de las muestras
3. Contraste ANOVA.

Tras analizar estadísticamente los resultados de ambas evaluaciones de usabilidad comparando los resultados de las preguntas y características entre la primera y segunda evaluación de usabilidad, el resultado generado por SPSS mostró un p de significación > 0.05 para todas las preguntas y

características analizadas, es decir se acepta la hipótesis nula (H_0 = las medias poblacionales son iguales). En otras palabras, no hubo ni mejoría ni tampoco empeoramiento de los elementos valorados de las evaluaciones de usabilidad según este análisis.

Capítulo 5: Evaluación de Usabilidad y Eficacia Educativa de GreedExCol

Uno de los objetivos principales de nuestro trabajo de tesis mencionado anteriormente fue la realización de una evaluación de usabilidad y eficacia educativa de nuestro enfoque educativo. La estructura de nuestro capítulo es la siguiente: el primer apartado describe la evaluación de usabilidad de la herramienta GreedExCol que soporta nuestro enfoque y el segundo apartado detalla la evaluación de eficiencia educativa.

5.1 Una Evaluación de Usabilidad de GreedExCol mediante Cuestionarios

Esta evaluación de GreedExCol se realizó en marzo de 2012 (Debdi, et al., 2012c), en la asignatura troncal “Diseño y Análisis de Algoritmos” de tercer curso de Ingeniería Informática. Las clases impartidas pertenecían al capítulo “Algoritmos voraces”. Los alumnos estaban familiarizados con los conceptos básicos de la técnica voraz, el método experimental y el asistente, dado que pudieron experimentar y familiarizarse con la herramienta antes de realizar la práctica obligatoria en el laboratorio donde experimentaron con el problema de la mochila que es fácil de descubrir sus funciones de selección óptimas mediante la experimentación. La práctica se realizó en dos sesiones de laboratorio de dos horas de duración cada una en un aula de informática. Al comienzo de la primera sesión de la práctica, los alumnos descargaron todo el material necesario: la práctica que contenía el enunciado del problema y una breve descripción de GreedExCol, la herramienta y una plantilla de informe que deben entregar. Los alumnos tenían que realizar varias tareas en grupos:

- 1) Utilizar GreedExCol para identificar las funciones de selección óptimas para el problema de la selección de actividades.
- 2) Llenar y presentar electrónicamente un breve informe escrito utilizando una plantilla de Word.

Después de esta sesión, el profesor dio dos horas de teoría adicional en clases en las que volvió a explicar entre otros, los conceptos de experimentación.

Al comienzo de la segunda sesión de práctica, el profesor identificó qué funciones de selección eran óptimas (OCF y ODC). Los alumnos tenían que trabajar en los mismos grupos de la primera sesión. Dependiendo de las respuestas dadas en la primera sesión, tenían que realizar tareas diferentes:

- Por cada función de selección identificada erróneamente como óptima, tenían que encontrar un contraejemplo.
- Por cada función de selección óptima que no habían identificado, tenían que revisar su experimentación y explicar por qué no la habían seleccionado documentando sus resultados en otro modelo de informe.

Tanto el cuestionario de usabilidad como el informe de la evaluación usados son los mismos que la evaluación de usabilidad de GreedEx descrita en el capítulo 4, excepto que en esta evaluación, el cuestionario se amplió con dos preguntas más sobre elementos concretos de GreedExCol (el menú de discusión: vistas comparativas, aceptar/rechazar propuestas) y una pregunta sobre la pestaña de tabla abreviada que no estaba incluida en el cuestionario anterior. El enunciado y el modelo de informe de la evaluación están en el Anexo 1, así como el cuestionario de usabilidad en el Anexo 2.

5.1.1 Valoraciones Numéricas sobre Aspectos Generales

En la Tabla 23 incluimos los resultados de las preguntas tipo *test* generales.

Tabla 23. Resultados numéricos de las preguntas generales (GreedExCol)

PREGUNTA	MEDIA	DESV. TÍPICA
Fácil de usar	4.11	1.10
Ha ayudado a analizar el efecto de cada estrategia voraz	4.07	0.90
Ha ayudado a identificar la estrategia óptima	4.32	0.90
Calidad general para analizar el efecto de cada estrategia voraz	3.79	0.99
En conjunto te ha gustado GreedExCol	3.71	0.71
Media total	4	

Destacan las respuestas obtenidas en la primera y tercera pregunta (facilidad de uso (4.11) y la identificación de la estrategia óptima (4.32)). El resultado más bajo se ha obtenido en las preguntas cuarta (calidad general) y quinta (satisfacción), con medias de 3.79 y 3.71, respectivamente.

Se agruparon las respuestas de las cinco preguntas por su puntuación recibida. Tal y como se puede ver en la Ilustración 17, los valores obtenidos han sido bastante altos. Las 140 respuestas se desglosan 102 altas o muy altas, 29 regulares y 9 malas o muy malas. Además, se observa que la moda de la pregunta de utilidad de GreedExCol (ha ayudado a identificar la estrategia óptima) y facilidad de uso es de 5 mientras que el resto de preguntas tienen una moda de 4.

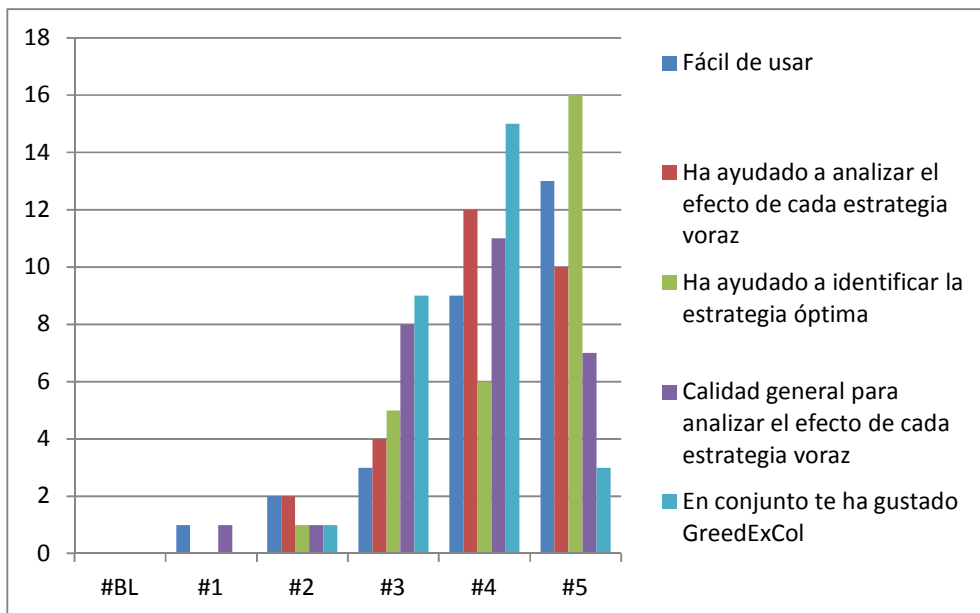


Ilustración 17. Valoración sobre aspectos generales de GreedExCol

5.1.2 Valoraciones Numéricas sobre Aspectos Concretos

La Tabla 24 muestra la evaluación de calidad de elementos concretos de GreedExCol, ordenados de mayor a menor.

Tabla 24. Resultados ordenados de la calidad de sus elementos (GreedExCol)

PREGUNTA	MEDIA	DESV. TÍPICA

Pestaña de tabla de datos de entrada.	3.79	1.03
Funciones de selección de estrategias	3.79	0.96
Pestaña de algoritmo	3.71	1.05
Pestaña de tabla abreviada	3.64	1.19
Exportación de imágenes y tablas	3.71	1.12
Panel de visualización	3.61	0.96
Pestaña de tabla de resumen	3.57	0.96
Funciones de Ejecución/Animación del algoritmo	3.57	1.00
Pestaña de problema	3.54	1.04
Menú discusión: vista comparativa de tus compañeros, aceptar/rechazar propuestos	3.61	1.03
Estructura del menú principal	3.43	1.10
Pestaña de tabla de resultados	3.39	1.07
Introducción o generación de datos de entrada	3.29	1.08
Iconos	3.25	1.14
Media total de respuestas	3.56	

Los resultados sobre la calidad de los elementos concretos son aceptablemente buenos. Como puede verse en la Tabla 24, el resultado medio de la evaluación es 3.56. De un total de 392 respuestas, 63 malas o muy malas, 122 regulares, 207 buenas o muy buenas y 0 en blanco. La mayoría de elementos concretos de GreedExCol tienen resultados cercanos, también podemos ver que la desviación típica no es muy grande, lo que nos confirma que los datos no son muy dispersos.

5.1.3 Críticas y Sugerencias Recibidas

Recurrimos a 4 preguntas de respuesta abierta para tener la opinión detallada de los alumnos sobre los aspectos positivos y negativos de GreedExCol. La Ilustración 18 contiene el número de respuestas en blanco y escritas para cada pregunta abierta.

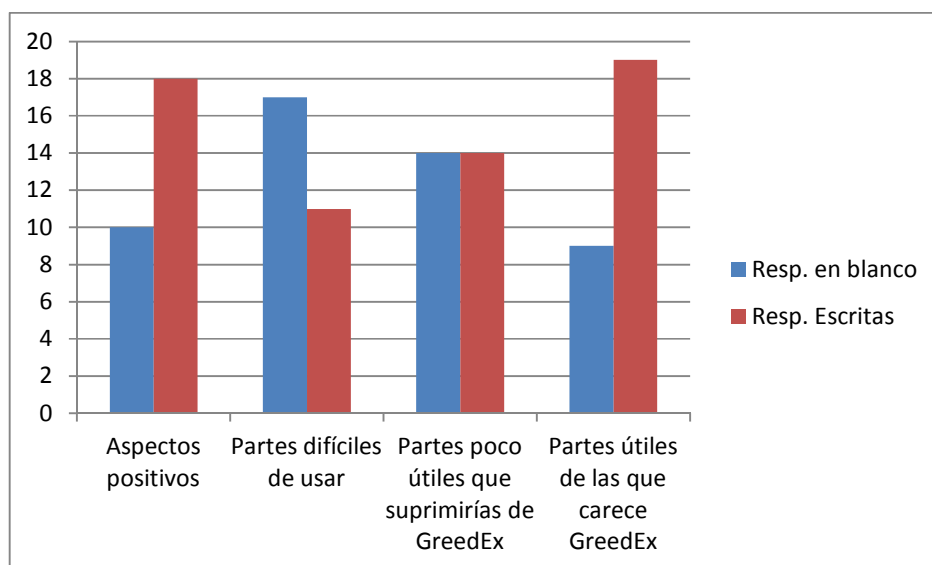


Ilustración 18. Preguntas de respuesta abierta sobre GreedExCol

En los aspectos positivos, se destacó su utilidad para hallar la estrategia óptima, la intuición de la aplicación, como la facilidad para aprender los algoritmos voraces. Algunos alumnos indicaron la utilidad de las tablas y la vista colaborativa. Otros factores también han sido identificados como la intuición de la parte de iconografía o la ventana de la discusión. Otros alumnos valoraron positivamente la generación aleatoria de datos de entrada, la ejecución de varias estrategias a la vez y la posibilidad de realizar muchas ejecuciones con su consecuente actualización de tablas que han sido útiles a la hora de encontrar la estrategia óptima mediante la comparación de resultados.

Con respecto a los partes más difíciles de usar, la mayoría de los alumnos no tenían comentarios que realizar. Aunque, algunos destacaron la parte de discusión como una de las partes más difíciles de usar, mientras otros señalaron la parte de las funciones de selección y la ausencia de un tutorial para el uso de GreedExCol.

En las partes de GreedExCol que el alumno suprimiría, los alumnos señalaron diferentes cuestiones como la pestaña de resultados, la exportación de tablas o imágenes y la parte de discusión.

Con respecto a las partes que sería útil incorporar, se destacó la mejora de la visualización de los resultados y la incorporación de la explicación breve de cada función de selección. Otros

sugirieron la posibilidad de tener una interfaz más dinámica para compartir. Por otra parte, un alumno sugirió la incorporación de un mensaje de notificación en la parte colaborativa de GreedExCol, para que el compañero sepa lo que quiere decir con la propuesta enviada. Otro propuso incluir una notificación más llamativa cuando algún compañero propone algún ejemplo y la adición de un apartado de *chat* para hacer más ágil el apartado de discusión. Otros propusieron mejorar la lentitud para cargar los datos desde la base de datos (que fue la mayor queja verbal durante las sesiones de la práctica). Por último, otros alumnos indicaron otros aspectos como:

- Añadir una opción que permita introducir el algoritmo.
- Guardar un fichero en forma de informe.
- Mostrar la complejidad.
- Portar GreedEx a la Web.
- Que se pueda ejecutar en Linux y no por consola.

5.1.4 Análisis de las Observaciones y los logs

Como mencionábamos anteriormente, durante las dos sesiones de evaluación ha asistido un observador que ha recogido las preguntas de los alumnos al profesor. Ha habido varios comentarios sobre la lentitud del programa, y sobre lo que hay que entregar en la práctica, pero también ha habido otras preguntas, que realizaron los alumnos, que muestrearon los dificultades encontradas por los alumnos durante la experimentación con GreedEx, como por ejemplo:

- No entender el tiempo total del problema de selección de actividades.
- No comprender el porqué de los resultados de las funciones de selección.
- No saber el significado de las funciones de selección.
- No poder detectar la función de selección óptima.
- Les resulta más fácil demostrar que las funciones de selección no son óptimas que demostrar lo contrario.

También hemos recogido otros tipos de preguntas que realizaron los alumnos como:

- Preguntas sobre la evidencia experimental en el informe.
- Preguntas sobre cómo se puede proponer un trabajo al compañero en la parte de discusión.

- Cómo exportar las imágenes.
- Posibilidad de borrar las ejecuciones anteriores.
- Problema encontrado a la hora de cargar los datos.

Por otra parte, durante la práctica, hemos estado registrando la interacción de los alumnos con el sistema GreedExCol mediante el uso de los logs. La

Tabla 25 muestra el porcentaje de alumnos que usaron algunos elementos de GreedExCol, ordenados de mayor a menor. En total, 30 alumnos usaron GreedExCol (información extraída de los logs).

Tabla 25. Porcentaje de alumnos que usaron algunos elementos de GreedExCol

ELEMENTOS CONSULTADOS GREDEXCOL LA SESIÓN	DE DURANTE	PORCENTAJE DE ALUMNOS QUE USARON EL ELEMENTO	PORCENTAJE DE ALUMNOS QUE NO USARON EL ELEMENTO
Tabla abreviada		100%	0%
Tabla de resultados globales		100%	0%
Tabla de resultados		96.67%	3.33%
Ejecución intensiva		96.67%	3.33%
Tabla de datos de entrada		83.33%	16.67%
Parte discusión		73.33%	26.67%
Activar el efecto de parpadeo		63.33%	36.67%
Pestaña de algoritmo		53.33%	46.67%
Exportar tablas		43.33%	56.67%
Pestaña de problema		40%	60%
Propuestas (parte colaborativa)		26.67%	73.33%
Ejecución paso a paso		26.67%	73.33%
Exportar imagen		20%	80%
Modificar datos de entrada		16.67%	83.33%

Según los logs recogidos, hemos podido hallar el porcentaje de alumnos que hayan usado algunos elementos concretos de GreedExCol, no hemos incluido todos de elementos porque su uso es muy esencial en la aplicación como por ejemplo (la ejecución completa, generación de datos,..)

Tal y como se puede observar, el porcentaje de alumnos que usaron la parte colaborativa es bastante alto, aunque ha habido pocos alumnos que hayan hecho propuestas a sus compañeros de

grupo. Otra cosa que puede observarse, es el porcentaje alto del uso de la ejecución intensiva, que sirvió de ayuda para detectar la función de selección óptima, aunque ha habido pocos alumnos que usaron la ejecución paso a paso.

Por otro lado, según las prácticas entregadas (9 grupos), todos los grupos han podido hallar las funciones de selección óptimas, aunque sólo dos de ellos identificaron además la función de selección de “Orden creciente de duración” como una función de selección óptima.

5.1.5 Conclusiones de la Evaluación de Usabilidad de GreedExCol

Los resultados obtenidos han sido positivos, tanto por la valoración de los alumnos como por la información recogida para mejorar GreedExCol. Los alumnos identificaron casi por igual los aspectos positivos de GreedExCol como los partes útiles de las que carece. En términos generales, lo más valorado de GreedExCol es su facilidad de uso y su ayuda a identificar las funciones de selección óptimas. Hay pocas funciones que suprimirían los alumnos y pocas las partes más difíciles que usar que suelen ser: la visualización de la parte de discusión y resultados.

Por otra parte, algunas dificultades detectadas en esta evaluación de usabilidad convencieron de la necesidad de revisar algunos elementos en GreedExCol como la lentitud en el acceso a la base de datos. Además se identificaron algunas mejoras para la parte de discusión como incluir una notificación más llamativa cuando algún compañero propone algún ejemplo, acompañar la propuesta con un mensaje y cambiar la visualización del apartado de discusión para poder ver los resultados de los compañeros con claridad. En resumen, esta evaluación sirvió para revisar algunos aspectos y poder soportar una buena parte de las sugerencias recogidas, las mejoras planteadas y corregir los problemas notificados.

5.2 Una Evaluación de Eficacia Educativa de GreedExCol

En esta sección se presenta la descripción y los resultados obtenidos de la evaluación de la eficacia educativa de GreedExCol. En la primera, segunda y tercera sección se describe el contexto educativo, diseño experimental y protocolo y tareas respectivamente. En la cuarta sección comentamos los resultados obtenidos y finalmente en la última sección incluimos las conclusiones.

5.2.1 Contexto Educativo

Esta evaluación de GreedExCol se realizó en abril de 2013 (Debdi, et al., 2013b), en la asignatura troncal “Diseño y Análisis de Algoritmos” de tercer curso de Ingeniería Informática. En el curso presentado se incluyó un capítulo sobre el análisis de la complejidad y de varios capítulos dedicados a las técnicas de diseño de algoritmos básicos (algoritmos voraces, dividir y vencerás y vuelta atrás).

Para evaluar la utilidad de GreedExCol se realizó un experimento para medir el nivel de conocimiento de los alumnos en dos grupos distintos, la asignación de los alumnos y los grupos fue independiente del experimento. La participación fue incentivada aumentando ligeramente la nota de la práctica sólo si la aprueban.

5.2.2 Diseño Experimental

La variable independiente fue el método de enseñanza utilizado para enseñar los algoritmos voraces. Los dos grupos recibieron un esquema experimental: planteamiento del problema, propuesta de funciones de selección y la experimentación para rechazar con contraejemplo u obtener pruebas de optimalidad. El grupo colaborativo recibió un esquema experimental y colaborativo utilizando el método experimental apoyado por GreedExCol mientras que el grupo de control usó otras herramientas de codificación.

La variable dependiente fue el incremento del nivel de conocimiento de los alumnos, medido como una comparación de las notas obtenidas por los alumnos antes y después del tratamiento. Ambos grupos llenaron una prueba previa (*pretest*) antes de recibir las clases de algoritmos voraces y una prueba posterior (*posttest*) justo después de la finalización de este capítulo. La prueba de conocimiento constaba de seis preguntas sobre los fundamentos de la optimización y algoritmos voraces, incluyendo teoría y problemas sencillos. Cada examen se puntuó en una escala que va de 0 (el grado más bajo) al 10 (el más alto). La prueba de conocimiento se puede encontrar en el Anexo 5.

La Tabla 26 muestra el número de alumnos que participaron en cada prueba, así como los alumnos que contestaron a las dos pruebas. Analizamos los resultados de este subgrupo de alumnos. El análisis se realizó con el programa SPSS 20.

Tabla 26. Alumnos participantes en la evaluación de eficacia educativa

	<i>PRETEST</i>	<i>POSTTEST</i>	<i>ALUMNOS COMUNES</i>
<i>Grupo de Control</i>	46	53	41
<i>Grupo Experimental/Colaborativo</i>	53	41	38

5.2.3 Protocolo y Tareas

Ambos grupos participaron en el mismo número de sesiones de clase y de laboratorio para el capítulo de algoritmos voraces. Todas de dos horas de duración. Las clases que se celebraron para este capítulo fueron iguales a excepción de dos. El capítulo de algoritmos voraces comenzó con una clase introductoria, seguida de dos sesiones que variaron en los dos grupos y finalmente se llevaron a cabo una serie de sesiones de clase donde se presentó una serie de algoritmos voraces conocidas (tales como Dijkstra, o de algoritmos de Kruskal y Prim).

La segunda sesión de clase difiere en la forma de consolidar los conocimientos de los alumnos en algoritmos voraces. Para los alumnos del grupo de control se les ilustró con problemas simples como el problema de cambio de monedas (Cormen, et al., 2001) o algunas variantes del problema de la mochila. Por otro lado, para los alumnos del grupo experimental, el profesor había utilizado GreedExCol en clase para dos problemas soportados por GreedExCol: maximizar el número de objetos en una mochila y maximizar el peso introducido en una mochila. Junto a los problemas, había presentado los conceptos de experimentación relacionados y el proceso de experimentación. Después, los alumnos pudieron experimentar en el aula informática con GreedExCol para el problema de la mochila. En la sesión de laboratorio, ambos grupos se les dio diferentes prácticas. Se pidió a los alumnos del grupo de control diseñar y programar un algoritmo voraz óptimo para el problema de selección de actividades. Por el otro lado, se pidió a los alumnos del grupo experimental aplicar el método experimental utilizando GreedExCol para el mismo problema de optimización, es decir, experimentar con las diferentes funciones de selección que ofrece la herramienta e identificar las óptimas.

5.2.4 Resultados Obtenidos

La Ilustración 19 muestra los diagramas de caja para ambos grupos. Con respecto al grupo de Control, el rango de notas está entre 2.5 y 5.25 en el *pretest* y entre 3.5 y 5 en el *posttest*. También

podemos observar un caso atípico en el *posttest*. Sin embargo, en el grupo experimental, el 50% de los alumnos sacaron entre 2.5 y 5 en el *pretest*, y entre 4 y 6.5 en el *posttest*.

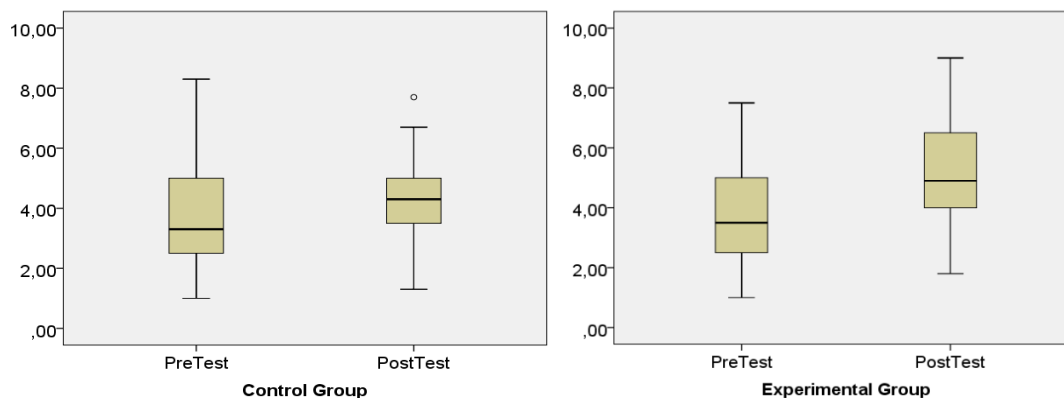


Ilustración 19. Diagramas de caja para los grupos analizados

Según el análisis descriptivo mostrado en la Tabla 27, hay una disminución de la dispersión de datos en el grupo de Control, sin embargo, el grupo experimental ha mostrado un aumento en la desviación típica. Por otra parte, la media del *posttest* es mayor que la media del *pretest* en ambos grupos. Para verificar que esta media es significativa respecto a su población, se compararon los resultados de ambas pruebas.

Tabla 27. Descriptivos de *pretest* y *posttest*

	PRETEST		POSTTEST	
	<i>Media</i>	<i>Desviación típica</i>	<i>Media</i>	<i>Desviación típica</i>
<i>Grupo de Control</i>	3.76	1.69	4.30	1.33
<i>Grupo Experimental</i>	3.72	1.46	5.18	1.83

El grupo de Control no sigue una distribución normal en la muestra del *pretest* (obteniendo $p < 0:05$ de significación de la prueba de *Shapiro-Wilk*), por eso, se ha optado por realizar la prueba no paramétrica de *Wilcoxon* en la que asumimos el rechazo de la hipótesis nula (obtención de $p < 0.05$ de significación), y concluimos que existen diferencias entre las notas de *pretest* y *posttest* para el grupo de Control, es decir, la media del *posttest* es estadísticamente significativa. En resumen, el nivel de conocimiento del grupo de Control había mejorado después de recibir las clases sobre algoritmos voraces. Por otra parte, dado que la muestra del grupo de Control no sigue una

distribución normalidad, se realizó una correlación de *Spearman* que demostró la existencia de una correlación significativa de 0.572 (obtención de $p < 0.05$ de significación) entre el *pretest* y el *posttest*.

Por otro lado, las muestras del grupo Colaborativo siguen una distribución normal (obtención de $p > 0:05$ de la significación usando la prueba de *Shapiro-Wilk* para ambas muestras). Por lo tanto, se realizó un *test* de *Levene* en el que se asumió la existencia de homogeneidad de las varianzas, y por lo tanto se realizó un contraste de ANOVA. En esta prueba se rechaza la hipótesis nula (obtención de $p < 0.05$ de significación), lo que implica que los resultados del *pretest* y el *posttest* son diferentes. Por lo tanto, podemos concluir que el uso de la herramienta GreedExCol mejora las calificaciones del *posttest*. Por una parte, dado que la muestra del grupo Colaborativo es normal, se realizó la prueba de correlación de *Pearson* que afirmó la existencia de una correlación significativa de 0.365 (obtención de $p < 0:05$ de significación) entre el *pretest* y el *posttest* del grupo Colaborativo. Por otra parte, debido a la normalidad de las dos muestras, se realizó una prueba *t Student* para muestras relacionadas que demostró la existencia de una diferencia significativa entre el *pretest* y el *posttest* para el grupo Colaborativo (obtención de $t = 4.766$ y $p = 0.00$ de significación) y una diferencia media de 1.457 entre el *pretest* y el *posttest*. Por lo tanto podemos concluir que hubo un aumento significativo en las notas del *posttest*. En resumen, hemos encontrado una fuerte evidencia de que GreedExCol mejora el rendimiento de los alumnos.

Con el fin de comparar las diferencias entre el aumento del nivel de conocimiento de los alumnos entre el grupo Colaborativo y de Control, se decidió calcular la magnitud o el tamaño del efecto, y por lo tanto obtener la significación de la diferencia encontrada entre ambos grupos. La Tabla 28 muestra un cambio de 0.34 para el grupo de Control que corresponde a un efecto pequeño (un efecto medio para $d = 0.5$) con un efecto de cambio de 0.17 y un porcentaje de aproximadamente 14%. Sin embargo, el grupo Colaborativo tuvo un valor *d de Cohen* de 0.87, lo que corresponde a un efecto grande (considerada grande para $d = 0.80$ o superior). Del mismo modo, un cambio porcentual de alrededor del 39% mostrando un nivel de cambio alto para el grupo Colaborativo.

Tabla 28. Cohen test

	COHEN'S D	EFFECT SIZE R	PORCENTAJE
<i>Grupo de Control</i>	0.34	0.17	14%
<i>Grupo Colaborativo</i>	0.87	0.40	39%

5.2.5 Conclusiones de la Evaluación de Eficiencia Educativa de GreedExCol

En conclusión, los resultados de esta evaluación muestran que el uso de la herramienta GreedExCol tiene una influencia positiva en el rendimiento de los alumnos, obteniendo mejoras significativas en las notas de *posttest* después de usar GreedExCol. Sin embargo, los alumnos que recibieron el esquema tradicional tenían una pequeña mejora en su desempeño. Esta afirmación se sustenta más aún cuando observamos que el resultado del tamaño del efecto (la cuantificación de la relevancia del efecto obtenido) es muy alto en el grupo Colaborativo por el uso de GreedExCol, y pequeño en el grupo de Control.

Capítulo 6: Evaluaciones de Motivación de GreedExCol

En este capítulo presentaremos una evaluación de motivación en diferentes paradigmas de aprendizaje con el objetivo de averiguar el efecto de GreedExCol en mejorar la motivación de los alumnos. La estructura del capítulo es la siguiente. En el primer apartado se presenta el cuestionario usado. En el segundo y el tercer apartado se presentan la primera y la segunda evaluación de motivación de GreedExCol. El apartado 4 muestra el análisis de la correlación entre la evaluación de usabilidad y la primera evaluación de motivación de GreedExCol. Finalmente el apartado 5 presenta el análisis de correlación entre la eficacia educativa de GreedExCol y la segunda evaluación de motivación.

6.1 Evaluación de Motivación de GreedExCol

Con el fin de evaluar la motivación situacional, Guay y sus colegas (Guay, et al., 2000) propusieron la Escala de Motivación Situacional (SIMS) que consta de 16 ítems que permiten la evaluación de las cuatro dimensiones de la motivación de acuerdo con la teoría de la autodeterminación: motivación intrínseca, la regulación identificada, regulación externa y desmotivación. Los autores de SIMS demostraron que la escala era adecuada para medir los diversos tipos de motivación en contextos educativos, tanto en el laboratorio como en estudios de campo, con niveles satisfactorios de consistencia interna en todas las subescalas de la motivación.

Martín- Albo y sus colegas (Martín-Albo, et al., 2009) han propuesto una versión más corta de SIMS, llamada EMSI (Escala de Motivación Situacional). Los autores evaluaron la escala EMSI en un contexto educativo para evaluar sus características psicométricas. Como consecuencia de ello, los autores propusieron eliminar los ítems 10 y 11 de la escala original SIMS (correspondiente a la regulación identificada y regulación externa, respectivamente) con el fin de mejorar la consistencia interna de las subescalas. Hemos elegido esta escala como un instrumento por su amplia utilización en el entorno educativo para evaluar la motivación situacional. Incluimos el cuestionario de motivación en el Anexo 3.

La escala EMSI resultante tiene 14 ítems agrupados en las cuatro dimensiones de la motivación. Cada elemento responde de manera diferente a la pregunta: "¿Por qué crees que debías realizar esta actividad?" Y se califica en una escala tipo Likert, que va del 1 (no se corresponde en absoluto con lo que pienso) a 7 (corresponde exactamente), con una puntuación intermedia de 4 (se corresponde exactamente con lo que pienso)

- La motivación intrínseca: ítems 1, 5, 9 y 11
- La motivación extrínseca a través de la regulación identificada: ítems 2, 6 y 12
- La motivación extrínseca a través de la regulación externa: ítems 3, 7 y 13
- La desmotivación: ítems 4, 8, 10 y 14

Se debe indicar que se ha realizado la inversa en las numeraciones de los alumnos en las preguntas 3, 4, 7, 8, 10 y 14 para realizar estos análisis usando el paquete estadístico SPSS 20.

6.2 Primera Evaluación de Motivación de GreedExCol

Esta evaluación de GreedExCol se realizó en marzo de 2012 (Debdi, et al., 2013f), en la asignatura troncal "Diseño y Análisis de Algoritmos". Los grupos participantes son:

1. Grupo Colaborativo que usó la herramienta GreedExCol.
2. Grupo Individual que usó la herramienta GreedEx.
3. Grupo de Control que recibió clases magistrales.

La evaluación se llevó a cabo en el mismo curso y año académico que la evaluación de usabilidad descrita en el apartado 5.1 Los grupos colaborativo e individual, así como las tareas que tenían que realizar también fueron los mismos como se describió anteriormente. Por otro lado, la tarea del grupo de control consistió en el diseño y la codificación de una función de selección óptima para el problema de selección de actividades. Los grupos participantes recibieron las mismas clases aunque se diferían en dos sesiones de las clases de los algoritmos voraces.

El objetivo de este experimento era medir la motivación en alumnos bajo diferentes paradigmas de aprendizaje. Los grupos completaron el cuestionario de motivación basada en la escala de EMSI después de asistir a las sesiones de laboratorio. Los cuestionarios se completaron de

forma anónima a través de una página web. El cuestionario de motivación usado se encuentra en el Anexo 3. La Tabla 29 recoge el número de alumnos participantes en cada grupo.

Tabla 29. Tamaño de las muestras de la primera evaluación de motivación

TIPO DE ACERCAMIENTO	NÚMERO DE PARTICIPANTES
Individual	62
Colaborativo	28
Control	17

6.2.1 Análisis de Motivación Global

Se han analizado estadísticamente los resultados del instrumento EMSI con sus 14 respuestas, así como las cuatro dimensiones, comparando los resultados entre los grupos experimentales (Individual y Colaborativo) y el grupo de Control. En total participaron 107 alumnos.

Según la Tabla 30, los grupos con mayor media son el grupo de Control e Individual. Para verificar que esta media es poblacionalmente significativa realizamos una comparación de los resultados del instrumento EMSI con un contraste de ANOVA sobre las medias poblacionales, realizando los siguientes pasos sobre los tres grupos:

1. Verificación de la normalidad de las muestras.
2. Verificación de la homogeneidad de las muestras o la prueba de *Kruskal Wallis* en el caso de la no normalidad de las muestras
3. Contraste ANOVA con *post-hoc* y corrección de *Bonferroni*.

Tabla 30. Descriptivos de los tres grupos participantes

	GRUPO DE CONTROL		GRUPO EXPERIMENTAL		GRUPO INDIVIDUAL	
	<i>Media</i>	<i>Desviación típica</i>	<i>Media</i>	<i>Desviación típica</i>	<i>Media</i>	<i>Desviación típica</i>
<i>Media motivación</i>	4.7053	0.62610	4.0539	0.75893	4.6560	0.60299

Para realizar una comparación de los resultados del instrumento EMSI con un contraste de ANOVA sobre las medias poblacionales, primero debemos saber si las muestras son normales. Y

según la prueba de *Kolmogorov-Sminov*, se concluye que el grupo Individual no sigue una distribución normal (obteniendo $p = 0.032 < 0.05$ de significación).

Dado que una de las muestras no cumplen la condición de normalidad, no podemos realizar un contraste de hipótesis ANOVA, pero sí la prueba de *Kruskal Wallis*, que es una prueba no paramétrica para muestras independientes (obteniendo $p=0.002$ de significación), por tanto rechazamos la hipótesis nula (no hay diferencias entre los tres grupos), es decir, existe como mínimo algún grupo que tiene diferencia con otro.

Una vez realizado el *test Kruskal Wallis* para todos los grupos, realizamos un análisis para contrastar cada tipo de experimento contra los demás para ver qué par de grupos es el que tiene diferencias.

En primer lugar, tras realizar el *test de Kruskal Wallis* para los grupos Colaborativo e Individual, se rechaza la hipótesis nula (obteniendo $p=0.00 < 0.05$ de significación), es decir, los grupos Colaborativo e Individual son diferentes.

En segundo lugar, tras realizar la prueba de *Kruskal Wallis* para los grupos Control e Individual, se acepta la hipótesis nula (obteniendo $p= 0.779 > 0.05$ de significación), es decir, los grupos de Control e Individual son iguales.

En tercer lugar, con respecto a la diferencia entre los grupos Colaborativo y Control, se rechaza la hipótesis nula según la prueba de *Kruskal Wallis* (obteniendo $p=0.014 < 0.05$ de significación), es decir, los grupos Colaborativo y Control son diferentes.

Interpretando todas las pruebas realizadas, podemos concluir que los grupos Control e Individual no presentan diferencia ninguna mientras el grupo Colaborativo es diferente a estos dos grupos. Por lo tanto, según estos últimos resultados y reinterpretando la Tabla 30 podemos afirmar que los grupos de Control e Individual mejoran la motivación con comparación con el grupo Colaborativo dada la lentitud de la herramienta GreedExCol.

6.2.2 Análisis de las Cuatro Dimensiones de la Motivación

Después de analizar la motivación en sus 4 dimensiones en general, ahora se analizan las motivaciones por cada una de las dimensiones. La Tabla 31 muestra los descriptivos de las dimensiones de motivación de cada grupo.

Tabla 31. Descriptivos de las dimensiones de motivación de para todos los grupos

	GRUPO DE CONTROL		GRUPO EXPERIMENTAL		GRUPO INDIVIDUAL	
	<i>Media</i>	<i>Desviación típica</i>	<i>Media</i>	<i>Desviación típica</i>	<i>Media</i>	<i>Desviación típica</i>
<i>Motivación Intrínseca</i>	3.98	1.23	3.47	0.88	4.08	0.80
<i>Regulación identificada</i>	5.01	1.12	4.00	1.22	4.71	0.91
<i>Regulación externa</i>	3.94	0.65	3.91	0.56	3.95	0.77
<i>Desmotivación</i>	5.76	0.79	4.77	1.44	5.70	1.09

6.2.2.1 Dimensión Motivación Intrínseca

Según la prueba de *Kolmogorov-Sminov*, la muestra del grupo Individual no sigue una distribución normal (obteniendo $p = 0.007 < 0.05$ de significación), sin embargo, el resto de muestras presenta normalidad.

Dado la no normalidad de una de las muestras, no podemos realizar un contraste de hipótesis ANOVA, pero sí la prueba no paramétrica de *Kruskal Wallis* para muestras independientes.

Interpretando las tablas generadas por el SPSS tras realizar la prueba de *Kruskal Wallis* para los tres grupos (obteniendo $p = 0.025 < 0.05$ de significación), podemos afirmar que existe al menos algún grupo diferente a otro. Por eso, optamos por realizar el *test* de *Kruskal Wallis* para cada par de grupos para determinar cuáles son los grupos diferentes.

En primer lugar, la prueba de *Kruskal Wallis* entre los grupos Colaborativo e Individual muestra una diferencia entre ambos grupos (obteniendo $p = 0.006 < 0.05$ de significación) rechazando la hipótesis nula ($H_0 =$ no hay diferencias entre los grupos). En segundo lugar, la prueba de *Kruskal Wallis* realizada para los grupos Colaborativo y Control muestra la no diferencia entre ambas muestras (obteniendo $p = 0.274 > 0.05$ de significación), aceptando la hipótesis nula ($H_0 =$ no

hay diferencias entre los grupos). En tercer lugar, la misma prueba realizada para los grupos Individual y Control mostró de nuevo la no diferencia de ambas muestras (obteniendo $p = 0.450 > 0.05$ de significación), aceptando la hipótesis nula ($H_0 =$ no hay diferencias entre los grupos).

Podemos concluir mediante la prueba de *Kruskal Wallis* realizada sobre cada par de grupos y reinterpretando la Tabla 31, que el grupo que usó la herramienta GreedEx mejora la motivación intrínseca que el grupo que usó GreedExCol. Sin embargo la motivación intrínseca no difiere en los grupos Colaborativo y Control.

6.2.2.2 Dimensión Regulación Identificada

Tras realizar la prueba de normalidad para los grupos en cuestión, se confirma la normalidad de dichas muestras (obteniendo $p > 0.05$ de significación), lo siguiente es averiguar la existencia de homogeneidad de varianzas para poder hacer un contraste de hipótesis ANOVA sobre las medias.

Para la verificación de la homogeneidad de varianzas se ha realizado el *test* de *Levene* (obteniendo $p = 0.244 > 0.05$ de significación), aceptando la hipótesis nula, es decir, existen homogeneidad de varianzas.

Una vez verificadas la normalidad y la homogeneidad de varianzas, podemos realizar un contraste de ANOVA (obteniendo $p = 0.03 < 0.05$ de significación), rechazándose la hipótesis nula ($H_0 =$ las medias poblacionales son iguales), lo que implica que al menos dos tipos de experimentos son diferentes.

Por otro lado, se realiza el *post-hoc Scheffé* de cada tipo de experimento contra los demás con el objetivo de ver las diferencias entre los grupos. Este *test* muestra diferencias significativas entre el grupo Colaborativo y Control (obteniendo $p = 0.009 < 0.05$ de significación) y los grupos Colaborativo e Individual (obteniendo $p = 0.014 < 0.05$ de significación). También se realizó la corrección de *Bonferroni* para evitar el error de múltiples contrastes simultáneos (corrección para múltiples comparaciones, donde el nuevo $p' = p/N$ tal que $N =$ número de *test*. En nuestro caso, $N = 3$, entonces $p' = 0.017$). Reinterpretando los valores de p de significación de los grupos analizados que son (0.014 y 0.009) con el nuevo $p' = 0.017$, que siguen siendo menores que p' (0.017) y por lo tanto, se confirman las diferencias entre el grupo Colaborativo y los dos grupos.

En resumen, según lo anterior y los resultados mostrados en la Tabla 31, podemos afirmar que los grupos Control e Individual mejoran la motivación en la dimensión regulación identificada que el grupo Colaborativo.

6.2.2.3 Dimensión Regulación Externa

Según la prueba de *Kolmogorov-Sminov*, no existe normalidad en el grupo Individual (obteniendo $p = 0.001 < 0.05$ de significación) en la dimensión Regulación Externa, aunque el resto de grupos muestra normalidad, por tanto no se puede hacer un contraste de hipótesis ANOVA sobre las medias.

Dado la no normalidad de una de las muestras, se opta por la prueba no paramétrica de *Kruskal-Wallis* para muestras independientes que confirma la no diferencia de muestras (obteniendo $p = 0.833 > 0.05$ de significación), es decir, no se rechaza la hipótesis nula, los grupos Colaborativo, Control e Individual no muestran una diferencia estadísticamente significativa entre sí.

6.2.2.4 Dimensión Desmotivación (*Amotivation*)

Tras realizar la prueba de normalidad, se concluye que el grupo de Control muestra normalidad, sin embargo los grupos Individual y Colaborativo no siguen una distribución normal (obteniendo $p < 0.05$ de significación en ambos grupos). Y según la prueba de *Kruskal Wallis* realizada sobre los tres grupos juntos, se rechaza la hipótesis nula (no hay diferencias entre los tres grupos), es decir, existe al menos un grupo como mínimo que muestra diferencia entre el resto de grupos (obteniendo $p = 0.012 < 0.05$ de significación). Lo siguiente es realizar un análisis para contrastar cada tipo de experimento contra los demás para ver qué par de experimentos tiene diferencias.

En primer lugar, la prueba de *Kruskal Wallis* llevada a cabo entre los grupos Colaborativo e individual muestra una diferencia entre ambas muestras (obteniendo $p = 0.005 < 0.05$ de significación), rechazando así la hipótesis nula (no hay diferencias entre los grupos Colaborativo e individual).

En segundo lugar, según la prueba de *Kruskal Wallis*, los grupos Colaborativo y Control son diferentes (obtención de $p = 0.025 < 0.05$ de significación) rechazando la hipótesis nula (no hay diferencias entre los grupos Colaborativo y Control)

En tercer lugar, la misma prueba realizada entre los grupos Control e Individual, nos confirma la no diferencia de muestras dado que el $p = 0.769 > 0.00$ de significación, por tanto no se rechaza la hipótesis nula, los grupos Control e Individual no son estadísticamente diferentes.

En resumen. Según las pruebas de *Kruskal Wallis* realizadas podemos afirmar que los grupos de Control e Individual están menos desmotivados que el grupo Colaborativo.

6.2.3 Conclusiones de la Primera Evaluación de Motivación

Los resultados de la evaluación de motivación realizada en diferentes paradigmas de aprendizaje son los siguientes:

- La motivación global se mejora en clases que usaron GreedEx o clases magistrales.
- Para las cuatro dimensiones:
 - Dimensión motivación Intrínseca: la motivación se mejora en los grupos que usaron GreedEx. Además no hubo diferencias entre los grupos Colaborativo y Control en esta dimensión.
 - Dimensión regulación identificada: la motivación en esta dimensión se mejora en los grupos que usaron GreedEx o clases magistrales.
 - Dimensión regulación externa: no hay diferencias entre los tres grupos analizados.
 - Dimensión de desmotivación: los grupos que usaron GreedEx y los que recibieron clases magistrales fueron menos desmotivados.

Se debe indicar que el decremento en la motivación en el grupo que usó la herramienta GreedExCol fue por el motivo de la lentitud de la herramienta dado que fue la principal crítica verbal de los alumnos el día de la práctica. Cabe mencionar que una nueva versión ha sido desarrollada solucionando este problema mediante las mejoras de las consultas SQL a la base de datos y la estructura de las tablas, además de incluir nuevas mejoras a la herramienta.

6.3 Segunda Evaluación de Motivación de GreedExCol

Esta evaluación de GreedExCol se realizó en abril de 2013 (Debdi, et al., 2013e) en la asignatura troncal “Diseño y Análisis de Algoritmos” en dos grupos diferentes: grupo colaborativo y control. En esta evaluación los dos grupos recibieron un esquema experimental: planteamiento del problema, propuesta de funciones de selección y la experimentación para rechazar con contraejemplo o obtener pruebas de optimalidad. El grupo colaborativo recibió un esquema experimental y colaborativo utilizando el método experimental apoyado por GreedExCol mientras que el grupo de control usó otras herramientas de codificación. El protocolo y las tareas que tenían que realizar los alumnos en los dos grupos fueron los mismos que la evaluación de eficacia educativa descrita en la sección 5.2.

Los grupos completaron una prueba de la motivación basada en la escala de EMSI antes y después de asistir a las sesiones de laboratorio. La Tabla 32 muestra el número de participantes en cada grupo. Todos los alumnos realizaron el *test* de manera anónima.

Tabla 32. Alumnos Participantes en la Motivación

GRUPOS	PRETEST	POSTTEST
<i>Grupo de Control</i>	53	47
<i>Grupo Colaborativo</i>	55	53

El objetivo de este experimento es medir la motivación en alumnos bajo dos diferentes paradigmas de aprendizaje. Se han hecho dos análisis al respecto, en primer lugar, se han analizado estadísticamente los resultados del instrumento EMSI con sus 14 respuestas, así como las cuatro dimensiones, comparando los resultados entre el grupo experimental y el grupo de control para los resultados de *pre* y *post* motivación. En segundo lugar, para determinar el efecto de GreedExCol en mejorar el nivel de motivación de los alumnos, se realizó un análisis *pre* y *post* motivación para cada grupo, para medir el nivel de crecimiento de la motivación. El cuestionario de motivación usado se encuentra en el Anexo 4.

6.3.1 Resultados del Análisis *pre* Motivación

En primer lugar, se realizó un análisis descriptivo de los grupos de Control y Colaborativo para los resultados de *pre* motivación que se muestra en la Tabla 33 y luego se ha llevado a cabo un análisis detallado entre ambos grupos.

Tabla 33. Descriptivos de *pretest*

	GRUPO DE CONTROL		GRUPO COLABORATIVO	
	<i>Media</i>	<i>Desviación típica</i>	<i>Media</i>	<i>Desviación típica</i>
Media Total	4.25	0.78	4.64	0.98
Motivación Intrínseca	3.33	1.09	3.76	1.26
Regulación identificada	4.71	1.13	5.07	1.30
Regulación externa	3.84	0.75	3.83	0.75
Desmotivación	5.13	1.35	5.80	1.39

Según la Tabla 33, el grupo con mayor media es el grupo colaborativo. Para verificar que esta media es poblacionalmente significativa, realizamos una comparación de los resultados del instrumento EMSI con un contraste ANOVA sobre las medias poblacionales, realizando los siguientes pasos sobre los dos grupos:

1. Verificación de la normalidad de las muestras.
2. Verificación de la homogeneidad de las muestras o la prueba de *Kruskal Wallis* en el caso de la no normalidad de las muestras.
3. Contraste ANOVA.

Para realizar una comparación de los resultados del instrumento EMSI con un contraste de ANOVA sobre las medias poblacionales, primero debemos saber si las muestras son normales. Para ello hemos utilizado la prueba de *Kolmogorov-Smirnov*. Se ha utilizado un error estándar del 5%.

6.3.1.1 Análisis de la Media de *pre* Motivación

Según la prueba de *Kolmogorov-Smirnov*, el grupo de Control sigue una distribución normal, sin embargo la muestra del grupo Colaborativo no lo sigue (obteniendo $p = 0.002 < 0.05$ de

significación). Por eso optamos por realizar el *test* de *Kruskal Wallis* para muestras independientes (obteniendo $p = 0.04 < 0.05$ de significación), donde rechazamos la hipótesis nula (H_0 = las medias poblacionales son iguales), es decir, los dos grupos son diferentes.

En resumen, podemos afirmar que el grupo Colaborativo que usó GreedExCol estaba más motivado.

6.3.1.2 Análisis de las Cuatro Dimensiones de Motivación

Después de analizar el *pre* motivación en sus 4 dimensiones en general, analizamos el *pre* motivación por cada una de sus dimensiones.

1) Dimensión Motivación Intrínseca

Tras realizar la prueba de *Kolmogorov-Smirnov* para los dos grupos analizados para la dimensión motivación intrínseca (obteniendo $p > 0.05$ de significación), se confirma la normalidad de las muestras. Lo siguiente es averiguar la existencia de homogeneidad de varianzas para poder hacer un contraste de hipótesis ANOVA sobre las medias.

Según la prueba de *Levene* (obteniendo $p = 0.194 > 0.05$ de significación), no se rechaza la hipótesis nula, es decir, existe homogeneidad de varianzas. Además, la prueba de ANOVA confirma la igualdad de las muestras (obteniendo $p = 0.060 > 0.05$ de significación), por tanto no se rechaza la hipótesis nula (H_0 = las medias poblacionales son iguales).

En resumen, no hay diferencias entre el grupo Colaborativo y el grupo de Control en la dimensión motivación intrínseca.

2) Dimensión Regulación Identificada

La prueba de *Kolmogorov-Smirnov* realizada para los grupos de Control y Colaborativo muestra la no normalidad del grupo Colaborativo (obteniendo $p = 0.033 < 0.05$ de significación). Por eso optamos por realizar la prueba de *Kruskal Wallis* para muestras no normales (obteniendo $p = 0.061 > 0.05$ de significación), lo cual confirma que no hay diferencias entre el grupo Colaborativo y Control en la dimensión regulación identificada.

3) Dimensión Regulación Externa

Según la prueba de *Kolmogorov-Smirnov* realizada en ambos grupos, se rechaza la hipótesis nula (H_0 = las muestras son normales) dado que $p < 0.05$ de significación, es decir, los grupos de Control y Colaborativo no son normales. Por lo tanto, no se puede realizar la prueba de ANOVA y optamos por realizar la prueba de *Kruskal Wallis* para muestras independientes.

Según la prueba de *Kruskal Wallis* (obteniendo $p = 0.785 > 0.05$ de significación), no se rechaza la hipótesis nula (H_0 = las medias poblacionales son iguales). Es decir, no hay diferencias entre el grupo Colaborativo y Control en la dimensión regulación externa.

4) Dimensión Desmotivación

Según la prueba de *Kolmogorov-Smirnov*, la muestra del grupo de Control es normal sin embargo la muestra del grupo Colaborativo no sigue una distribución normal dado que el $p = 0.000 < 0.05$ de significación. Por tanto, no podemos aplicar el contraste de ANOVA en la dimensión de desmotivación. En este caso, para la comparación de las muestras optamos por usar la prueba no paramétrica de *Kruskal Wallis*, rechazando la hipótesis nula (H_0 = las medias poblacionales son iguales), obteniendo $p = 0.004 < 0.05$ de significación, lo que implica que los dos grupos presentan diferencias.

Y según la Tabla 33, podemos afirmar que el grupo Colaborativo tiene mejorías en la desmotivación, es decir los alumnos se encuentran menos desmotivados dado que se trata de datos invertidos.

6.3.2 Resultados del Análisis de *Post* Motivación

Se han analizado estadísticamente los resultados del instrumento EMSI con sus 14 respuestas, así como las cuatro dimensiones, comparando los resultados entre el grupo Colaborativo y el grupo de Control para los resultados de *post* motivación.

Tabla 34. Descriptivos de *posttest*

	<i>GRUPO DE CONTROL</i>	<i>GRUPO EXPERIMENTAL</i>

	<i>Media</i>	<i>Desviación típica</i>	<i>Media</i>	<i>Desviación típica</i>
<i>Media total</i>	4.10	0.85	4.90	0.83
<i>Motivación Intrínseca</i>	3.21	1.15	4.60	1.03
<i>Regulación identificada</i>	4.46	1.37	5.25	1.05
<i>Regulación externa</i>	3.73	0.69	3.92	0.79
<i>Desmotivación</i>	5.00	1.48	5.67	1.43

Según la Tabla 34 el grupo con mayor media es el grupo Colaborativo. Para verificar que esta media es poblacionalmente significativa realizamos una comparación de los resultados del instrumento EMSI con un contraste de ANOVA sobre las medias poblacionales, realizando los siguientes pasos sobre los tres grupos:

4. Verificación de la normalidad de las muestras.
5. Verificación de la homogeneidad de las muestras o la prueba de *Kruskal Wallis* en el caso de la no normalidad de las muestras
6. Contraste ANOVA.

6.3.2.1 Análisis de la Media Global de *post* Motivación

Dado la no normalidad de ambas muestras (obteniendo $p = 0.001 < 0.05$ de significación de la prueba de *Kolmogorov-Smirnov* para el grupo Colaborativo y obteniendo $p = 0.041 < 0.05$ de significación de *Shapiro Wilk* para el grupo de Control), no se puede hacer un análisis ANOVA. Por eso se ha optado por realizar la prueba no paramétrica de *Kruskal Wallis* (obteniendo $p = 0.00 < 0.05$ de significación) donde se rechazó la hipótesis nula, es decir, las dos muestras son diferentes. Y según las Tabla 34, podemos afirmar que la motivación se mejora en el grupo Colaborativo.

6.3.2.2 Análisis de las Cuatro Dimensiones de la Motivación

Después de analizar la motivación en sus 4 dimensiones en general, ahora se analizan las cuatro dimensiones de la *post* motivación.

1) Dimensión Motivación Intrínseca

Las pruebas de normalidad para los dos grupos muestran valores de $p > 0.05$ de significación, entonces no se rechaza la hipótesis nula (H_0 = las muestras son normales), las muestras son normales. Y según la prueba de *Levene*, no se rechaza la hipótesis nula (obteniendo $p = 0.360 > 0.05$ de significación), es decir, existe homogeneidad de varianzas. Una vez verificadas la normalidad y la homogeneidad de varianzas se realiza el contraste de ANOVA (obteniendo $p = 0.00 < 0.05$ de significación) rechazando la hipótesis nula (H_0 = las medias poblacionales son iguales), lo que implica las dos muestras son diferentes. Y según la Tabla 34, podemos afirmar que la motivación intrínseca se mejora en el grupo Colaborativo.

2) Dimensión Regulación Identificada

La prueba de normalidad realizada a ambos grupos indica la no normalidad de la muestra del grupo Colaborativo (obteniendo $p = 0.002$ de significación de *Kolmogorov Smirnov*), lo siguiente es realizar la prueba de *Kruskal Wallis* para muestras independientes (obteniendo $p = 0.002 < 0.05$ de significación) que rechaza la hipótesis nula (H_0 = las medias poblacionales son iguales), lo que implica que las dos muestras son diferentes. En resumen, podemos afirmar que la dimensión de la regulación identificada se mejora en el grupo Colaborativo.

3) Dimensión Regulación Externa

En esta dimensión, no se pudo realizar un análisis ANOVA dado que la muestra del grupo Colaborativo no es normal (obteniendo $p = 0.045 < 0.05$ de significación de la prueba de *Kolmogorov Smirnov*). Por eso se ha optado por realizar la prueba no paramétrica de *Kruskal Wallis* (obteniendo $p = 0.317 > 0.05$ de significación), lo que implica que las muestras no son diferentes. Es decir, no existe ninguna diferencia entre los dos grupos analizados en la dimensión regulación externa.

4) Dimensión Desmotivación

Según la prueba de *Shapiro Wilk* para el grupo de Control (obteniendo $p = 0.006$ de significación) y la prueba de normalidad para el grupo Colaborativo (obteniendo $p = 0.006$ de significación de *Kolmogorov Smirnov*), se rechaza la hipótesis nula, es decir, no existe normalidad en ambos grupos. Por eso se ha optado por realizar la prueba no paramétrica de *Kruskal Wallis*, que según el resultado de ésta, se rechaza la hipótesis nula (obteniendo $p = 0.009 < 0.05$ de

significación), lo que implica que las dos muestras son diferentes. Es decir, el grupo Colaborativo tiene mejorías en la desmotivación, en otras palabras, los alumnos del grupo Colaborativo se encuentran menos desmotivados ya que se trata de datos invertidos.

6.3.4 Análisis *pre-post* del grupo de Control

En este apartado se presentan los resultados del instrumento EMSI con sus 14 respuestas, así como las cuatro dimensiones de motivación, comparando los resultados del *pre* y *post* motivación del grupo de Control.

6.3.4.1 Análisis de la Media de Motivación

La Tabla 35 muestra los valores descriptivos de las dos muestras del grupo de Control.

Tabla 35. Valores descriptivos de motivación global (grupo de control)

	NÚMERO DE ALUMNOS	MEDIA	DESVIACIÓN TÍPICA
<i>Pre</i>	47	4.223	0.745
<i>Post</i>	47	4.106	0.857

Según la Tabla 35, la media del *pre* motivación es mayor que la media del *post* motivación. Para verificar que esta media es poblacionalmente significativa realizamos una comparación de los resultados de los dos *test* con un contraste de ANOVA sobre las medias poblacionales, realizando los siguientes pasos:

1. Verificación de la normalidad de las muestras.
2. Verificación de la homogeneidad de las muestras o la prueba de *Wilcoxon* en el caso de la no normalidad de las muestras.
3. Contraste ANOVA.

Según la prueba de normalidad de *Shapiro Wilk*, no existe normalidad en la muestra de *post* Motivación (obteniendo $p = 0.041$ de significación). Por eso se ha optado por realizar la prueba de *Wilcoxon* (obteniendo $p = 0.173$ de significación), es decir, se acepta la hipótesis nula ($H_0 =$ las

medias poblacionales son iguales). Por tanto, no existen diferencias entre el *pre* y *post* motivación del grupo de control en la media total de motivación.

6.3.4.2 Análisis de las Cuatro Dimensiones de Motivación

Después de analizar la motivación en sus 4 dimensiones en general, ahora se analiza las cuatro dimensiones de motivación para el grupo de control.

1) Dimensión Motivación Intrínseca

La Tabla 36 muestra los valores descriptivos de las dos muestras del grupo de Control.

Tabla 36. Valores descriptivos de motivación intrínseca (grupo de control)

	MEDIA	DESVIACIÓN TÍPICA
<i>Pre</i>	3.271	1.064
<i>Post</i>	3.218	1.159

La prueba de normalidad realizada para los grupos de Control y Colaborativo muestra la normalidad de ambas muestras dado que $p > 0.05$ de significación, aceptando la hipótesis nula (las muestras son normales), lo siguiente es averiguar la existencia de homogeneidad de varianzas, y según la prueba de *Levene*, existe homogeneidad de varianzas (obteniendo $p = 0.405 > 0.05$ de significación).

Por otra parte, según la prueba de ANOVA realizada, se acepta la hipótesis nula (obteniendo $p = 0.817 > 0.05$ de significación), lo que implica que el *pre* y *post* motivación son iguales en la dimensión motivación intrínseca para el grupo de Control.

2) Dimensión Regulación Identificada

La Tabla 37 muestra los valores descriptivos de las dos muestras del grupo de Control.

Tabla 37. Valores descriptivos de regulación identificada (grupo de control)

	MEDIA	DESVIACIÓN TÍPICA
<i>Pre</i>	4.666	1.135

<i>Post</i>	4.468	1.370
-------------	--------------	-------

Según la prueba de normalidad realizada sobre las dos muestras analizadas, se acepta la hipótesis nula (las muestras son normales), obteniendo un valor de sig > 0.05 de significación, lo siguiente es averiguar la existencia de homogeneidad de varianzas mediante la prueba de *Levene*, confirmando así la existencia de la homogeneidad de varianzas (obteniendo $p = 0.130 > 0.05$ de significación).

Una vez verificadas la normalidad y la homogeneidad de varianzas realizamos un contraste ANOVA (obteniendo $p = 0.446 > 0.05$ de significación), por lo que se acepta la hipótesis nula ($H_0 =$ las medias poblacionales son iguales), lo que implica que las dos muestras de *pre* y *post* motivación del grupo de Control no son diferentes.

3) Dimensión Regulación Externa

La Tabla 38 muestra los valores descriptivos de las dos muestras del grupo de Control.

Tabla 38. Valores descriptivos de regulación externa (grupo de control)

	MEDIA	DESVIACIÓN TÍPICA
<i>pre</i>	3.836	0.779
<i>post</i>	3.737	0.698

La prueba de normalidad realizada sobre la muestra de la *pre* motivación muestra la no normalidad de la muestra (obteniendo $p = 0.007$ de significación de la prueba de *Shapiro Wilk*), entonces se rechaza la hipótesis nula ($H_0 =$ las muestras son normales). Por tanto realizamos la prueba de *Wilcoxon* para muestras relacionadas (obteniendo $p = 0.553 > 0.05$ de significación), confirmando la igualdad de muestras, es decir, las muestras de *pre* y *post* motivación en la dimensión regulación externa no son diferentes.

4) Dimensión Desmotivación

La Tabla 39 muestra los valores descriptivos de las dos muestras del grupo de Control.

Tabla 39. Valores descriptivos de desmotivación (grupo de control)

	MEDIA	DESVIACIÓN TÍPICA
<i>pre</i>	5.133	1.318
<i>post</i>	5.00	1.480

Según la prueba de *Shapiro Wilk*, no existe normalidad en la muestra de *post* motivación (obteniendo $p = 0.006 < 0.05$ de significación), por tanto realizamos la prueba de *Wilcoxon* para muestras relacionadas que según ésta, se acepta la hipótesis nula ($H_0 =$ las medias poblacionales son iguales), obteniendo $p = 0.231 > 0.05$ de significación, es decir, las muestras de *pre* y *post* motivación del grupo de Control son iguales en la dimensión desmotivación.

6.3.5 Análisis *pre-post* del grupo Colaborativo

En este apartado se presentan los resultados del instrumento EMSI con sus 14 respuestas, así como las cuatro dimensiones, comparando los resultados del *pre* y *post* motivación del grupo Colaborativo.

6.3.5.1 Análisis de la Media de Motivación

La Tabla 40 muestra los valores descriptivos de las dos muestras del grupo Colaborativo.

Tabla 40. Valores descriptivos de motivación (grupo colaborativo)

	NÚMERO DE ALUMNOS	MEDIA	DESVIACIÓN TÍPICA
<i>pre</i>	44	4.634	1.039
<i>post</i>	44	4.913	0.845

Según la prueba de *Shapiro Wilk*, no existe normalidad en ambas muestras, (obteniendo $p = 0.000$ de significación para la muestra de *pre* motivación y obteniendo $p = 0.016$ de significación para la muestra de *post* motivación). Por tanto realizamos la prueba de *Wilcoxon* para muestras relacionadas, lo cual, se acepta la hipótesis nula (obteniendo $p = 0.293$ de significación), por tanto las muestras de *pre* y *post* motivación son iguales.

6.3.5.2 Análisis de las cuatro dimensiones

Después de analizar la motivación en sus 4 dimensiones en general, ahora se analiza las motivaciones para cada una de las dimensiones para el grupo colaborativo.

1) Dimensión Motivación Intrínseca

La Tabla 41 muestra los valores descriptivos de las dos muestras del grupo Colaborativo.

Tabla 41. Valores descriptivos de motivación intrínseca (grupo colaborativo)

	MEDIA	DESVIACIÓN TÍPICA
<i>Pre</i>	3.789	1,339
<i>Post</i>	4.630	1.053

Según la prueba de *Shapiro Wilk*, existe la normalidad en las dos muestras (obteniendo $p > 0.05$ de significación), aceptando la hipótesis nula (H_0 =las muestras son normales), lo siguiente es averiguar la existencia de homogeneidad de varianzas. Y según la prueba de *Levene*, se acepta la hipótesis nula (obteniendo $p = 0.071 > 0.05$ de significación), es decir, existen homogeneidad de varianzas.

Una vez verificadas la normalidad y la homogeneidad de varianzas, podemos realizar un contraste de ANOVA que a su vez muestra la no igualdad de las muestras (obteniendo $p = 0.002 < 0.05$ de significación). Y según la Tabla 41, podemos afirmar que la motivación se mejora en la *post* motivación en la dimensión motivación intrínseca.

2) Dimensión Regulación Identificada

La Tabla 42 muestra los valores descriptivos de las dos muestras del grupo Colaborativo.

Tabla 42. Valores descriptivos de regulación identificada (grupo colaborativo)

	MEDIA	DESVIACIÓN TÍPICA
<i>Pre</i>	5.090	1.366
<i>Post</i>	5.204	1.116

Según la prueba de *Shapiro Wilk*, no existe normalidad en ambas muestras (obteniendo $p = 0.004 < 0.05$ de significación) para la muestra de *pre* motivación y (obteniendo $p = 0.012 < 0.05$ de significación) para la muestra de *post* motivación, rechazando la hipótesis nula (H_0 = las muestras son normales). Por tanto realizamos la prueba de *Wilcoxon* para muestras relacionadas que muestra un valor de $p = 0.799 > 0.05$ de significación, que indica que las muestras de *pre* y *post* motivación son iguales en la dimensión regulación identificada.

3) Dimensión Regulación Externa

La Tabla 43 muestra los valores descriptivos de las dos muestras del grupo Colaborativo.

Tabla 43. Valores descriptivos de regulación externa (grupo colaborativo)

	MEDIA	DESVIACIÓN TÍPICA
<i>Pre</i>	3.803	0.771
<i>Post</i>	4.022	0.814

Según la prueba de *Shapiro Wilk*, existe normalidad en las muestras analizadas para las muestras de *pre* y *post* motivación, lo siguiente es averiguar la existencia de homogeneidad de varianzas mediante la prueba de *Levene*, confirmando así la existencia de la homogeneidad de varianzas (obteniendo $p = 0.590 > 0.05$ de significación).

Una vez verificadas la normalidad y la homogeneidad de varianzas realizamos un contraste ANOVA (obteniendo $p = 0.198 > 0.05$ de significación), lo que implica que las muestras de *pre* y *post* motivación son iguales en la dimensión regulación externa.

4) Dimensión desmotivación

La Tabla 44 muestra los valores descriptivos de las dos muestras del grupo Colaborativo.

Tabla 44. Valores descriptivos de desmotivación (grupo colaborativo)

	MEDIA	DESVIACIÓN TÍPICA
<i>Pre</i>	5.761	1.477
<i>Post</i>	5.647	1.392

Según la prueba de *Shapiro Wilk*, no existe normalidad en ambas muestras (obteniendo $p = 0.000$ de significación). Por tanto, realizamos la prueba de *Wilcoxon* para muestras relacionadas (obteniendo $p = 0.487 > 0.05$ de significación), aceptando la hipótesis nula ($H_0 =$ las medias poblacionales son iguales), es decir, las muestras de *pre* y *post* motivación son iguales en la dimensión desmotivación.

6.3.6 Conclusiones de la Segunda Evaluación de Motivación

Con respecto a los resultados de *pre* motivación, no existe diferencias entre los dos grupos, excepto en la dimensión de la desmotivación de tal manera que el grupo Colaborativo fue menos desmotivado que el grupo de Control.

En cuanto a los resultados de *post* motivación, la motivación global se mejoró en el grupo Colaborativo. Por otra parte, con respecto al análisis dimensional de *post* motivación, encontramos:

- Dimensión de motivación intrínseca: la motivación se mejoró en el grupo Colaborativo.
- Dimensión de regulación identificada: esta dimensión se mejoró en el grupo Colaborativo.
- Dimensión de regulación externa: no hubo diferencias entre los dos grupos en esta dimensión.
- Dimensión de Desmotivación: el grupo Colaborativo fue menos desmotivado.

Con el fin de medir el aumento de nivel de motivación en ambos grupos, se realizó un análisis de motivación *pre* versus *post* motivación en cada grupo a través del contraste de ANOVA para muestras normales o el *test* de *Wilcoxon* en caso de la no normalidad de muestras, en estas pruebas, se asumió que no había diferencias entre la *pre* y *post* motivación para el grupo de Control, sin embargo, la motivación intrínseca ha mejorado después de usar la herramienta GreedExCol para el grupo Colaborativo.

6.4 Correlación entre Usabilidad y Motivación de GreedExCol

Las evaluaciones de usabilidad y motivación se realizaron en marzo de 2012 en la asignatura troncal “Diseño y Análisis de Algoritmos” del curso de Grado de Ingeniería del Software. Después se ha optado por realizar un análisis de correlación entre ambas (Debdi, et al., 2013c). El protocolo y

las tareas se encuentran en el apartado 5.1 para la evaluación de usabilidad y el apartado 6.2 para la evaluación de motivación.

6.4.1 Resultados de la Correlación entre Motivación y Usabilidad

Presentamos los resultados del análisis de correlación entre la motivación y la usabilidad de GreedExCol. Recordemos que participaron 27 alumnos comunes en ambas evaluaciones. Hemos dividido las preguntas del cuestionario de usabilidad en tres secciones:

1. Aspectos visuales:

- Los iconos (*Iconos*)
- Visualización del gráfico (*Panel V*)
- Pestaña de problema (*P. problema*)
- Pestaña de algoritmo (*P. Algoritmo*)
- Estructura del menú (*Estructura*)
- Tabla de datos de entrada (*T. Entrada*)
- Tabla histórica (*T. Abreviada*)
- Tabla de resultados (*T. Resultados*)
- Tabla de resumen (*T. Resumen*)

2. Aspectos funcionales:

- Fácil de usar (*Facilidad*)
- Exportar imágenes y tablas (*Exportación*)
- Selección de funciones de selección (*SelecciónEstrategias*)
- Utilidad para analizar el efecto de las funciones de selección (*Analizar*)
- Utilidad para identificar la estrategia óptima (*Identificar*)
- Facilidad y buena funcionalidad para analizar las estrategias voraces (*Calidad*)
- Aceptación de GreedExCol en general (*TGustaGreedExCol*)

3. Otros aspectos:

- Facilidad y funcionalidad para introducir y generar datos de entrada (*Introducción/GeneraciónDatos*)

- Facilidad y buena funcionalidad para realizar ejecuciones del algoritmo y animaciones (*Ejecución/Animación*)
- Calidad de los servicios de discusión en grupo (*Menú Discusión*)

6.4.1.1 Correlación entre la Media de Motivación y Usabilidad

Con el objetivo de realizar la prueba de *Pearson*, primero se comprobó la normalidad de todas las muestras analizadas (la media de motivación y las variables de usabilidad). Para ello, hemos utilizado la prueba de normalidad de *Kolmogorov-Smirnov* que confirmó la existencia de normalidad (obtención de sig > 0.05 de significación) en todas las variables de usabilidad y la muestra de la motivación global (calculada a partir de las preguntas invertidas).

La Tabla 45 muestra la correlación de *Pearson* entre la *media de motivación* y las variables de usabilidad de GreedExCol con carácter visual.

Tabla 45. Correlación de aspectos visuales

CORRELACIÓN	MEDIA MOTIVACIÓN	SIG (PRUEBA DE SIGNIFICACIÓN)
Estructura	0.241	0.224
Iconos	0.288	0.145
Panel Visualización	0.310	0.115
P. Problema	0.189	0.343
P. Algoritmo	0.350	0.073
T. Entrada	0.504	0.007*
T. Resultados	0.643	0.00*
T. Resumen	0.580	0.002*
T. Abreviada	0.420	0.029 **
* Correlación significativa al nivel 0.01		
** Correlación significativa al nivel 0.05		

Las celdas resaltadas en negrita de la Tabla 45 muestran una correlación positiva entre la *tabla de entrada*, *tabla de resultados*, *tabla de resumen* y *tabla de abreviada* y la *media de motivación*.

La Tabla 46 muestra la correlación de *Pearson* entre la *media de motivación* y las variables de usabilidad de GreedExCol con carácter funcional.

Tabla 46. Correlación de aspectos funcionales

CORRELACIÓN	MEDIA MOTIVACIÓN	SIG (PRUEBA DE SIGNIFICACIÓN)
Selección Estrategias (<i>SelecciónEstrategias</i>)	0.358	0.066
Identificar la estrategia óptima (<i>Identificar</i>)	0.232	0.244
Analizar las estrategias voraces (<i>Calidad</i>)	0.576	0.002 *
Facilidad	0.438	0.022 **
Exportación	0.523	0.005 *
Útil para analizar el efecto de las estrategias (<i>Analizar</i>)	0.637	0.00 *
Aceptación de GreedExCol en general	0.467	0.014 **
* Correlación significativa al nivel 0.01 ** Correlación significativa al nivel 0.05		

Tal y como está mostrado en la Tabla 46, existe una correlación positiva entre la *media motivación* y la *facilidad de uso, exportación de imágenes y tablas, la utilidad para analizar el efecto de las estrategias, el análisis de las estrategias voraces y la aceptación de GreedExCol*.

La Tabla 47 muestra la correlación entre la motivación global y el resto de variables de usabilidad de GreedExCol.

Tabla 47. Correlación de otros aspectos

CORRELACIÓN	MEDIA MOTIVACIÓN	SIG (PRUEBA DE SIGNIFICACIÓN)
Introducción/Generación datos	0.224	0.262
Ejecución/Animación	0.298	0.131
Menú Discusión	0.062	0.757

Según la prueba de correlación de *Pearson* mostrada en la Tabla 47, no existe ninguna correlación significativa entre el resto de variables de usabilidad de GreedExCol y la media de motivación.

En resumen, vemos que el resultado es indeterminado porque hay variables que sí tienen una cierta relación y otras no. Esto da que pensar que puede haber varias tendencias dentro de las dimensiones de la motivación, con lo cual vamos a hacer un análisis por dimensiones.

6.4.1.2 Correlación entre las Cuatro Dimensiones de Motivación y Usabilidad

Recordamos que para los estudios por dimensiones se han mantenido las puntuaciones originales. Según la prueba de *Kolmogorov-Smirnov*, existe normalidad en las muestras de motivación en sus cuatro dimensiones (obtención de $\text{sig} > 0.05$ de significación).

La Tabla 48 muestra la correlación entre las cuatro dimensiones de motivación y las variables de usabilidad con aspecto visual.

Tabla 48. Correlación de aspectos visuales

CORRELACIÓN	INTRÍNSECA	REGULACIÓN IDENTIFICADA	REGULACIÓN EXTERNA	DESMOTIVACIÓN
Estructura	0.147 (sig=0.466)	0.24897892 (sig=0.210)	-0.261 (sig=0.188)	-0.189 (sig=0.344)
Iconos	0.152 (sig= =0.452)	0.433** (sig=0.024)	-0.323 (sig=0.100)	-0.166 (sig=0.407)
Panel Visualización	0.086 (sig= 0,670)	0.278 (sig=0.159)	-0.238 (sig=0.230)	-0.350 (sig=0.073)
P. Problema	-0.0003 (sig = 0.998)	-0.036 (sig=0.858)	-0.035 (sig=0.862)	-0.341 (sig=0.082)
P. Algoritmo	0.137 (sig= 0.494)	0.152 (sig=0,448)	-0.039 (sig=0,844)	-0.378 (sig=0.052)
T. Entrada	0.177 (sig= 0.376)	0.093 (sig=0.642)	-0.138 (sig=0.491)	-0.624* (sig=0.001)
T. Resultados	0.369 (sig= 0.058)	0.295 (sig=0.135)	-0.271 (sig=0.171)	-0.597* (sig=0.001)
T. Resumen	0.269 (sig= 0.173)	0.332 (sig=0.091)	-0.214 (sig=0.282)	-0.547* (sig=0.003)
T. Abreviada	0.161 (sig=0.421)	0.296 (sig=0.133)	-0.233 (sig=0.241)	-0.408** (sig=0.034)
* Correlación significativa al nivel 0.01 ** Correlación significativa al nivel 0.05				

En la Tabla 48, vemos varias correlaciones entre las cuatro dimensiones y los variables de usabilidad de GreedExCol con carácter visual:

- Correlación positiva entre la dimensión *regulación identificada* y los *iconos*.
- Correlación negativa entre la dimensión *desmotivación* y la *tabla de entrada*, *tabla de resultados*, *tabla de resumen* y *tabla abreviada*.

La Tabla 49 muestra la correlación de *pearson* entre las cuatro dimensiones de motivación y las variables de usabilidad de GreedExCol con carácter funcional.

Tabla 49. Correlación de aspectos funcionales

CORRELACIÓN	INTRÍNSECA	REGULACIÓN IDENTIFICADA	REGULACIÓN EXTERNA	DESMOTIVACIÓN
Facilidad	0.290 (sig =0.142)	0.138 (sig=0.491)	-0.251 (sig=0.207)	-0.476** (sig=0.012)
Exportación	0.375 (sig =0.054)	0.314 (sig=0.110)	-0.3660012 (sig=0.060)	-0.356 (sig=0.068)
Selección Estrategias	0.213 (sig =0.284)	0.107 (sig=0.595)	-0.371 (sig=0.056)	-0.335 (sig=0.087)
Útil para analizar el efecto de las estrategias (Analizar)	0.458 (sig =0.016 (correlación significativa al nivel 0.05))	0.34474093 (sig=0.078)	-0.2222 (sig=0.265)	-0.579* (sig=0.002)
Identificar la estrategia óptima (Identificar)	0.2442 (sig =0.219)	-0.0941 (sig=0.641)	-0.4379** (sig=0.022)	-0.326 (sig=0.097)
Analizar las estrategias voraces (Calidad)	0.413** (sig =0.032)	0.396** (sig=0.041)	-0.0691 (sig=0.732)	-0.4937* (sig=0.009)
Aceptación de GreedExCol en general	0.3138 (sig =0.111)	0.3165 (sig=0.108)	0.095 (sig=0.634)	-0.383** (sig=0.048)
* Correlación significativa al nivel 0.01 ** Correlación significativa al nivel 0.05				

Según los resultados de la correlación de *pearson* podemos afirmar la existencia de las siguientes correlaciones:

- Correlación positiva entre la dimensión *motivación intrínseca* y la *utilidad para analizar el efecto de las estrategias* y el *análisis de las estrategias voraces*.
- Correlación positiva entre la dimensión *regulación identificada* y el *análisis de las estrategias voraces*.
- Correlación negativa entre la dimensión *regulación externa* y la *identificación de la estrategia óptima*.

- Correlación negativa entre la dimensión *desmotivación y facilidad de uso, utilidad para analizar el efecto de las estrategias, el análisis de las estrategias voraces y la aceptación de GreedExCol en general.*

La Tabla 50 muestra la correlación entre las cuatro dimensiones de motivación y el resto de variables de usabilidad de GreedExCol.

Tabla 50. Correlación de otros aspectos

CORRELACIÓN	INTRÍNSECA	REGULACIÓN IDENTIFICADA	REGULACIÓN EXTERNA	DESMOTIVACIÓN
Introducción/Generación de datos	0.188 (sig=0.349)	0.223 (sig=0.263)	-0.317 (sig=0.107)	-0.190 (sig=0.343)
Ejecución/Animación	0.124 (sig=0.536)	0.245 (sig=0.218)	-0.263 (sig=0.186)	-0.280 (sig=0.157)
Menú Discusión	0.221 (sig=0,267)	-0.316 (sig=0.109)	0.072 (sig=0.720)	-0.180(sig=0.370)

Según la Tabla 50, no hay ninguna correlación entre las cuatro dimensiones de motivación y el resto de variables de usabilidad.

6.4.2 Conclusiones de la Correlación entre Usabilidad y Motivación

Según los resultados de la correlación realizada sobre los elementos de GreedExCol y la motivación de los alumnos, podemos ver que hay variables de la interfaz de usuario que sí están correlacionadas con la variable *Media Motivación* y otras no, tanto variables que miden aspectos visuales de la interfaz (*Estructura, Iconos, Panel de visualización, etc.*) como las que miden aspectos funcionales (*Exportar imágenes y tablas, Funciones de selección, Útil para analizar el efecto de las estrategias, etc.*). Por tanto podemos afirmar que hay partes de la interfaz que sí están relacionadas con la motivación.

En primer lugar, las variables de aspectos visuales de la interfaz *T. Entrada, T. Resultados, T. Resumen* y *T. Abreviada* están correlacionadas con la variable *Media Motivación*. Estas variables hacen referencia a los widgets *Table* que muestran información de los datos de entrada y los resultados de las ejecuciones del algoritmo voraz para cada una de las estrategias. Por tanto la usabilidad de los widgets *Table* de la UI está relacionada con la motivación del alumno. Esta

afirmación se sustenta más aún cuando observamos que estas mismas variables están correlacionadas inversamente con la dimensión Desmotivación. Es decir, que los alumnos que percibieron estos widgets *Table* con menor usabilidad estuvieron más desmotivados para la realización de la tarea. Por tanto, los widgets *Table* están relacionados con la motivación del alumno positivamente para la realización de las tareas en el proceso de aprendizaje de algoritmos voraces. El motivo de esto puede ser que este tipo de controles ofrece la información de forma muy clara y precisa. Además, estos controles de GreedExCol proporcionan abundante información sobre las funciones de selección, sus efectos, solución que proporcionan, resumen de varias ejecuciones, etc. Todos estos elementos son percibidos por el alumno como aspectos muy positivos y valorados que le ayudan a determinar la función de selección óptima del algoritmo voraz. Probablemente esto genera en el alumno una sensación de seguridad y control sobre la tarea, por lo que puede ser que se motive más que otro alumno que no tiene esta percepción.

En segundo lugar, los variables de aspectos funcionales de la interfaz, por ejemplo, la variable *Facilidad* de la UI tiene una correlación directa con la variable *Media Motivación*. Además, tiene una correlación inversa con la dimensión de la *Desmotivación*. Por tanto podemos señalar que la facilidad de uso y utilización de la UI están relacionados con la motivación que tiene el alumno en el proceso de aprendizaje de los algoritmos voraces. Esta influencia en la motivación es más fuerte aun cuando el alumno percibe que la herramienta es una ayuda para analizar el efecto que tienen las funciones de selección para el problema voraz en concreto con el que está trabajando. Esto lo observamos por la correlación directa que tiene la variable de usabilidad *Útil para analizar el efecto de las estrategias* con las variables de motivación *Media Motivación* y *Motivación Intrínseca* y por la correlación inversa que tiene con la *Desmotivación*. Esto significa que los alumnos que perciben la herramienta como una ayuda para resolver el problema voraz que se les planteó en concreto se motivan tanto que usan la herramienta únicamente por el placer de usarla. Éste es el grado de motivación más valorado que puede experimentar el usuario.

En tercer lugar, la barra de herramientas está relacionada con la motivación. Hay una correlación directa entre la variable *Iconos* y la variable de la dimensión *Regulación identificada*. Esto significa que los alumnos perciben la barra de herramientas como un elemento importante de la interfaz si bien no les gusta demasiado. En principio no encontramos una explicación sobre esta cuestión.

En cuarto lugar, el resto de las partes de la interfaz no tiene ninguna relación con la motivación del alumno durante el proceso de aprendizaje de algoritmos voraces. Esto es sorprendente, especialmente el *widget* que muestra gráficamente la animación de comportamiento del algoritmo para las diferentes estrategias, ya que en principio al ser gráfico, parecería que podría ser motivante para el alumno, y sin embargo no lo es. Esto puede ser debido a que la información que visualiza gráficamente este *widget* sobre las estrategias voraces es visualizada en forma de texto por los *widgets Table* y estos últimos *widgets* sí que presentan una relación con la motivación del alumno tal y como hemos expuesto más arriba. Además las visualizaciones del problema voraz utilizado fueron simples y sencillas, presentando éstas poco valor añadido al alumno respecto a la información presentada en los *widgets Table*.

6.5 Correlación entre Motivación y Eficiencia de GreedExCol

Las evaluaciones de motivación y eficiencia educativa se realizaron en Abril de 2013 en la asignatura troncal “Diseño y Análisis de Algoritmos” del curso de Grado de Ingeniería del Software en dos grupos diferentes (Debdi, et al., 2013a). El protocolo y las tareas se encuentran en el apartado 6.3 para la evaluación de motivación y el apartado 5.2 para la evaluación de eficiencia educativa.

La Tabla 51 muestra el número de alumnos participantes comunes en cada grupo. Este análisis se realizó con el programa paquete estadístico SPSS 20.

Tabla 51. Alumnos participantes comunes en la correlación

GRUPOS	ALUMNOS COMUNES
<i>Grupo de Control</i>	30
<i>Grupo Colaborativo</i>	31
<i>Total</i>	61

6.5.1 Resultados de la Correlación entre Motivación y Eficiencia Educativa

Presentamos los resultados de la correlación entre el *pretest* y *posttest* de conocimiento y los resultados de *pre* y *post* motivación en sus cuatro dimensiones para los grupos de control, colaborativo y los dos grupos juntos respectivamente.

En primer lugar, con respecto al grupo de control, según la prueba de *Kolmogorov-Smirnov*, los *pretest* y *posttest* de conocimiento, *pre* y *post* motivación en sus cuatro dimensiones siguen una distribución normal, por lo que se realiza la correlación de *Pearson*. Según éste, existe una única correlación negativa entre la desmotivación del *post* motivación y el *pretest* de conocimiento para el grupo de control que carece de significado.

En segundo lugar, con respecto al grupo colaborativo, la prueba de *Kolmogorov-Smirnov* para los *test* de conocimiento, los *test* de motivación en sus cuatro dimensiones afirma la normalidad de estas muestras, por lo que se realiza la correlación de *pearson* que confirma la no existencia de ninguna correlación significativa entre las notas del *pretest* y *posttest* de conocimiento y de motivación para el grupo colaborativo.

Por último, con respecto los dos grupos juntos, según la prueba de *Kolmogorov-Smirnov*, los *test* de conocimiento y de motivación en sus cuatro dimensiones siguen una distribución normal, entonces se realiza la correlación de *pearson*. Según éste, se puede afirmar la no existencia de ninguna correlación significativa entre los *test* de conocimiento y de motivación en sus cuatro dimensiones para los dos grupos juntos.

6.5.2 Resultados de la Correlación entre Aumento de Motivación y Aumento de Conocimiento

Presentamos los resultados de la correlación entre la eficiencia educativa o el aumento de conocimiento medida como la resta de las calificaciones obtenidas por los alumnos antes y después del tratamiento (*posttest- pretest*) y el aumento de la motivación en sus cuatro dimensiones calculado a partir de la resta de la motivación obtenida antes y después del tratamiento (*posttest- pretest*).

Con respecto al grupo de control, según la prueba de *Kolmogorov-Smirnov*, las muestras del aumento de la dimensión intrínseca, regulación identificada, desmotivación, la media de motivación y el aumento de conocimiento siguen una distribución normal, lo siguiente es averiguar la correlación de *pearson* que confirma que no existe ninguna correlación entre la eficiencia y las muestras de motivación analizadas. Y según la correlación de *spearman* entre el aumento de la regulación externa que no sigue una distribución normal y la eficiencia educativa, tampoco existe una correlación entre

ambas muestras. En resumen, no hay ninguna correlación entre el aumento de conocimiento y de motivación para el grupo de control.

Con respecto al grupo colaborativo, según la prueba de *Kolmogorov-Smirnov*, las muestras del aumento de las dimensiones de motivación intrínseca, regulación externa, desmotivación y aumento de conocimientos siguen una distribución normal, así que se realiza la correlación de *pearson* que demuestra que no existe ninguna correlación significativa entre el aumento de conocimiento y las muestras de motivación analizadas para el grupo colaborativo. Y según la correlación de *spearman* entre el aumento de conocimiento y el resto de muestras del aumento de motivación que no siguen una distribución normal, no existe ninguna correlación entre el aumento de conocimiento y la dimensión regulación identificada, ni con la media de la motivación. En resumen, no hay ninguna correlación entre el aumento de conocimiento y de motivación de los alumnos para el grupo colaborativo.

6.5.2 Conclusiones de la Correlación entre Eficiencia y Motivación

Hemos presentado de forma detallada un análisis de correlación entre eficiencia y motivación en sus cuatro dimensiones para dos grupos que recibieron diferentes paradigmas de aprendizaje, habiéndose obtenido ninguna correlación significativa entre el aumento de motivación y de conocimiento en ambos grupos. Tampoco hubo ninguna correlación significativa entre los test de conocimiento y de motivación en sus cuatro dimensiones por separado ni en ambos grupos.

Capítulo 7: Análisis de Estilos de Aprendizaje en Alumnos de Informática

En este capítulo presentamos el análisis de estilos de aprendizaje en alumnos de informática incluyendo los análisis de correlación entre los estilos de aprendizaje y el aumento de conocimiento y el aumento de motivación.

7.1 Análisis de Estilos de Aprendizaje

El cuestionario de estilos de aprendizaje de Felder y Silverman (Felder & Silverman, 1988) es un modelo que tiene como objetivo captar las diferencias de estilo de aprendizaje más importantes entre los alumnos de ingeniería y así proporcionar una buena base para los profesores para diseñar un método de enseñanza que aborde las necesidades de aprendizaje de todos alumnos. Este cuestionario proporciona:

- Las preferencias de aprendizaje de los individuos o de clases.
- Las fortalezas y debilidades en un ambiente académico.

La evaluación de estilos de aprendizaje se realizó en abril de 2013 en la asignatura troncal “Diseño y Análisis de Algoritmos” del curso de Grado de Ingeniería del Software (Debdi, et al., 2014a). El objetivo del experimento era medir los estilos de aprendizaje en los alumnos de ingeniería. Este modelo consiste en 4 dimensiones, cada una con dos extremos. La Tabla 52 muestra los ocho estilos de aprendizaje.

Tabla 52. Estilos de aprendizaje

Activo. Tienden a probar nuevas cosas, prefieren trabajar en grupo.	Reflexivo. Prefieren trabajar solos.
Visual. Imágenes, diagramas, diagramas de flujo.	Verbal. Requiere explicaciones escritas y verbales.
Sensorial. Cosas concretas, hechos y procedimiento.	Intuitivo. Aprende conceptos.
Global. Pensamiento holístico, saltos grandes.	Secuencial. Pensamiento lineal, pequeños pasos.

Para conseguir ésta medición han participado dos grupos de Ingeniería de software de la Universidad Rey Juan Carlos: grupo Colaborativo que usó la herramienta GreedExCol y otro de Control bajo un esquema tradicional. El protocolo usado está descrito en el apartado 5.2. El cuestionario de estilos de aprendizaje se encuentra en el Anexo 6. La Tabla 53 muestra el número de alumnos participantes en cada grupo para cada estilo de aprendizaje.

Tabla 53. Alumnos participantes en la evaluación de estilos de aprendizaje

Estilos de aprendizaje	Grupo de Control	Grupo Colaborativo
Total	44	49

7.1.1 Resultados del Análisis de Estilos de Aprendizaje para Todos los Grupos

La Tabla 54 muestra el número de alumnos por cada uno de los ocho estilos de aprendizaje para todos los alumnos. En total fueron 110 alumnos.

Tabla 54. Distribución de los alumnos según los estilos de aprendizaje

Estilos de aprendizaje	Alumnos
Activo	85
Reflexivo	25
Sensorial	88
Intuitivo	22
Visual	95
Verbal	15
Secuencial	78
Global	32

Como puede verse en la Ilustración 20, existe un porcentaje alto de alumnos cuyo estilo de aprendizaje es (activo, sensorial, visual y secuencial) frente al (reflexivo, intuitivo, verbal y global).

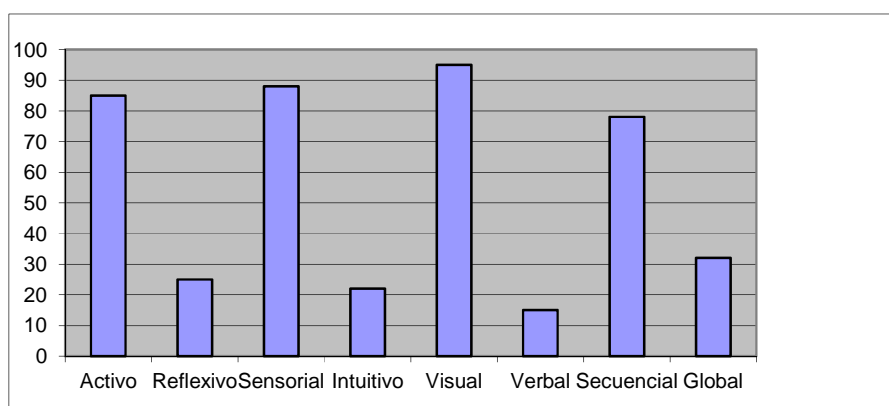
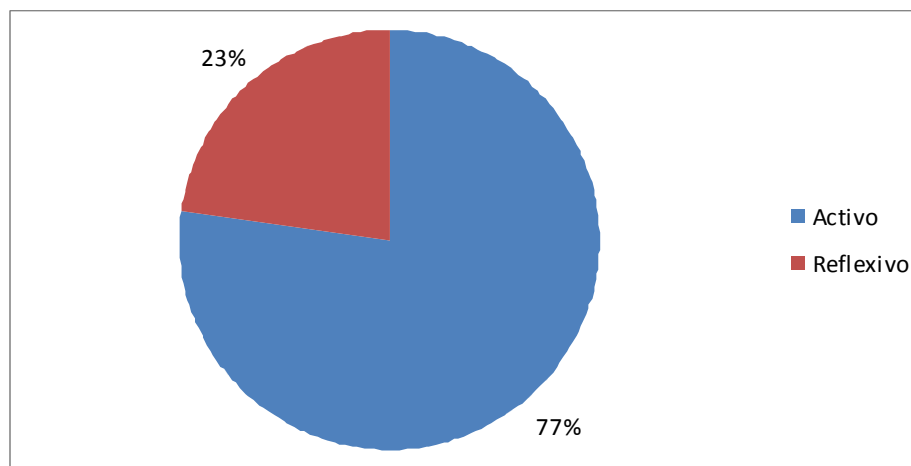


Ilustración 20. Estilos de aprendizaje de los dos grupos**7.1.1.1 Estilo Activo/Reflexivo**

Los procesos mentales que convierten la información percibida en conocimiento constan de dos categorías: la experimentación activa y la observación reflexiva. La experimentación activa implica hacer algo con la información en el mundo exterior, como hablar de ello, mientras que la observación reflexiva implica examinar y manipular la información introspectivamente ((Felder & Silverman, 1988), p. 678).

Los alumnos cuyo estilo de aprendizaje es activo no aprenden mucho de las conferencias o clases tradicionales, sin embargo aprenden mejor experimentando y trabajando en grupo. Por otro lado, los alumnos cuyo estilo es reflexivo requieren situaciones que dan oportunidad para pensar acerca de la información que se presenta ((Felder & Silverman, 1988), p.679).

La Ilustración 21 muestra el porcentaje del total de alumnos cuyo estilo de aprendizaje es activo frente al reflexivo, como puede verse, hay más alumnos activos que reflexivos.

**Ilustración 21.** Porcentaje de alumnos Activos/Reflexivos**7.1.1.2 Estilo Sensorial/Intuitivo**

La percepción y la intuición son dos formas en las que la gente tiende a percibir el mundo. La percepción implica la observación y recogida de datos a través de los sentidos, la intuición implica la

percepción indirecta a través del inconsciente: la especulación, la imaginación y corazonadas. Aunque los alumnos usan tanto estas facultades, la mayoría prefiere el uso de una u otra ((Felder & Silverman, 1988), p. 676).

Los alumnos cuyo estilo es sensorial son buenos para memorizar hechos y tienden a ser cuidadosos y lentos en la realización de su trabajo, mientras que los intuitivos prefieren principios, teorías e innovación, pero la repetición no les gusta. Los detalles pueden aburrirlos, son buenos en captar nuevos conceptos y tienden a completar las tareas rápidamente ((Felder & Silverman, 1988), p. 676).

Según la Ilustración 22, el 20% es el porcentaje del total de alumnos cuyo estilo de aprendizaje es intuitivo frente al 80% que es sensorial.

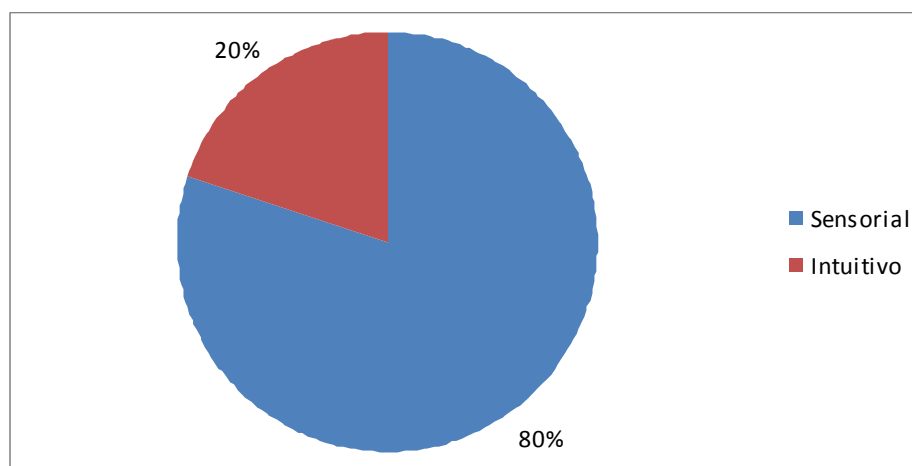


Ilustración 22. Porcentaje de alumnos Sensoriales/Intuitivos

7.1.1.3 Estilo Visual/Verbal

Los alumnos visuales recuerdan mejor lo que ven: imágenes, diagramas, diagramas de flujo, líneas de tiempo, películas, demostraciones. Se les puede olvidar la información que se les comunica verbalmente. Sin embargo, los alumnos cuyo estilo de aprendizaje es verbal, recuerdan mucho de lo que oyen y aún más de lo que se ve y se oye (cuando la información es presentada tanto visual como verbalmente). Ellos recuerdan y aprenden bien de las charlas, prefieren una explicación verbal a la

demostración visual y aprenden con eficacia explicando las cosas a los demás ((Felder & Silverman, 1988), p. 677).

Según la Ilustración 23, el porcentaje del total de alumnos cuyo estilo de aprendizaje es visual es el 86% frente al 14% cuyo estilo de aprendizaje es verbal.

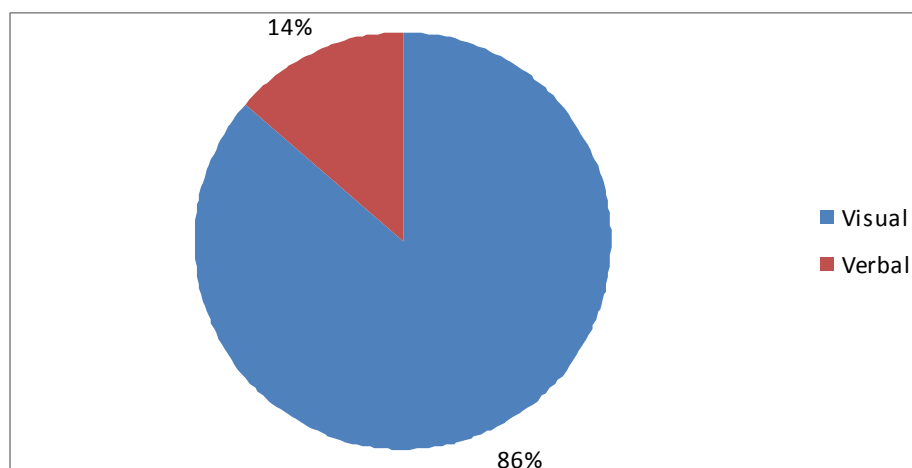


Ilustración 23. Porcentaje de alumnos Visuales/Verbales

7.1.1.4 Estilo Secuencial/Global

Los alumnos cuyo estilo de aprendizaje es secuencial se sienten cómodos con el material presentado en el dominio de una progresión ordenada lógicamente, siguiendo los procesos de razonamiento lineal en la resolución de problemas. Pueden ser fuertes en el pensamiento y el análisis convergente y aprenden mejor cuando los profesores presentan material en una progresión constante de dificultad ((Felder & Silverman, 1988), p. 679). Sin embargo, los alumnos de estilo global tienden a aprender de forma fragmentada: se pueden sentir perdidos por días o semanas sin poder resolver problemas simples o mostrar la comprensión más elemental, hasta que de pronto "hacen las cosas", pueden tener dificultades para trabajar con material que sólo tiene una comprensión parcial o superficial. Ellos tienden a hacer saltos intuitivos y tienen dificultades para explicar cómo llegan a la solución ((Felder & Silverman, 1988), p. 679).

La Ilustración 24 muestra el porcentaje total de alumnos cuyo estilo de aprendizaje es secuencial frente al global. Podemos ver claramente que la mayoría de los alumnos son más secuenciales que globales.

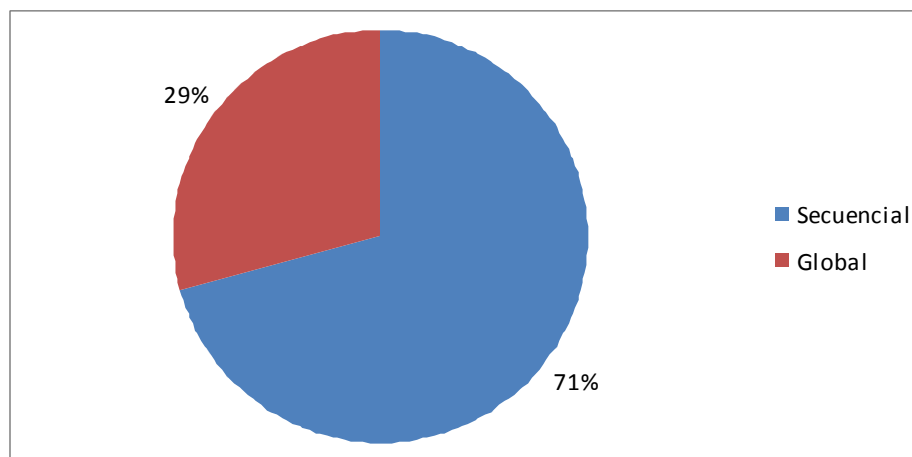


Ilustración 24. Porcentaje de alumnos Secuenciales/Globales

7.1.2 Resultados del Análisis de Estilos de Aprendizaje para el Grupo de Control

La Tabla 55 muestra el porcentaje de cada uno de los estilos de aprendizaje para el grupo de alumnos que recibió un esquema tradicional. En total participaron 44 alumnos.

Tabla 55. Estilos de aprendizaje para el grupo de control

Estilos de aprendizaje	Alumnos	Porcentaje
Sensorial	36	82%
Intuitivo	8	18%
Visual	38	86%
Verbal	6	14%
Secuencial	32	73%
Global	12	27%
Activo	33	75%
Reflexivo	11	25%

Según la Tabla 55 y la Ilustración 25, podemos concluir que los alumnos que recibieron un esquema tradicional son más sensoriales, visuales, secuenciales y activos.

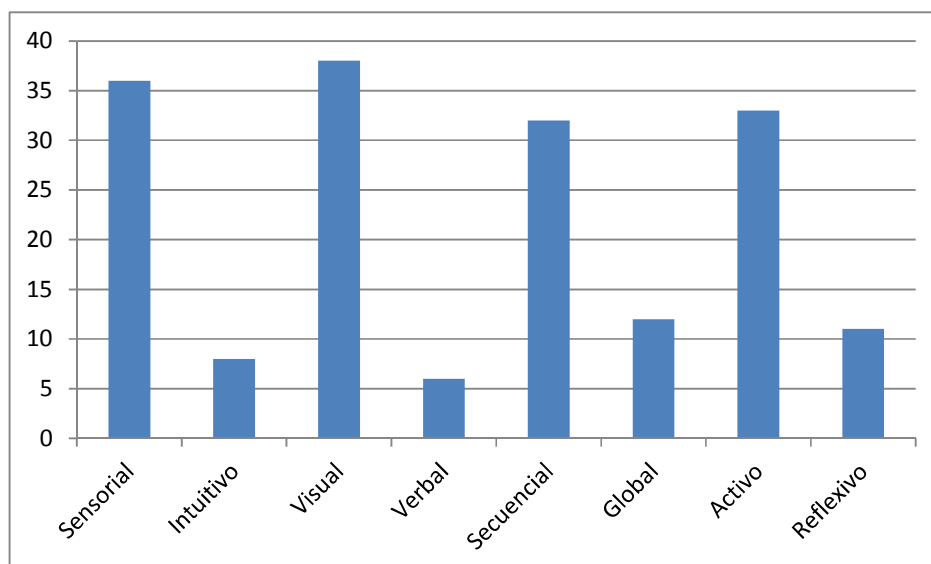


Ilustración 25. Estilos de aprendizaje para el grupo de Control

7.1.3 Resultados del Análisis de Estilos de Aprendizaje para el Grupo Colaborativo

La Tabla 56 muestra el porcentaje de estilos de aprendizaje de los alumnos del grupo colaborativo (que utilizaron la herramienta colaborativa GreedExCol). En total participaron 49 alumnos.

Tabla 56. Estilos de aprendizaje para el grupo colaborativo

Estilos de aprendizaje	Alumnos	Porcentaje
Sensorial	39	80%
Intuitivo	10	20%
Visual	43	88%
Verbal	6	12%
Secuencial	33	67%
Global	16	33%
Activo	37	76%
Reflexivo	12	24%

Según la Tabla 56 y la Ilustración 26, podemos concluir que los alumnos del grupo Experimental/Colaborativo son más sensoriales, visuales, secuenciales y activos.

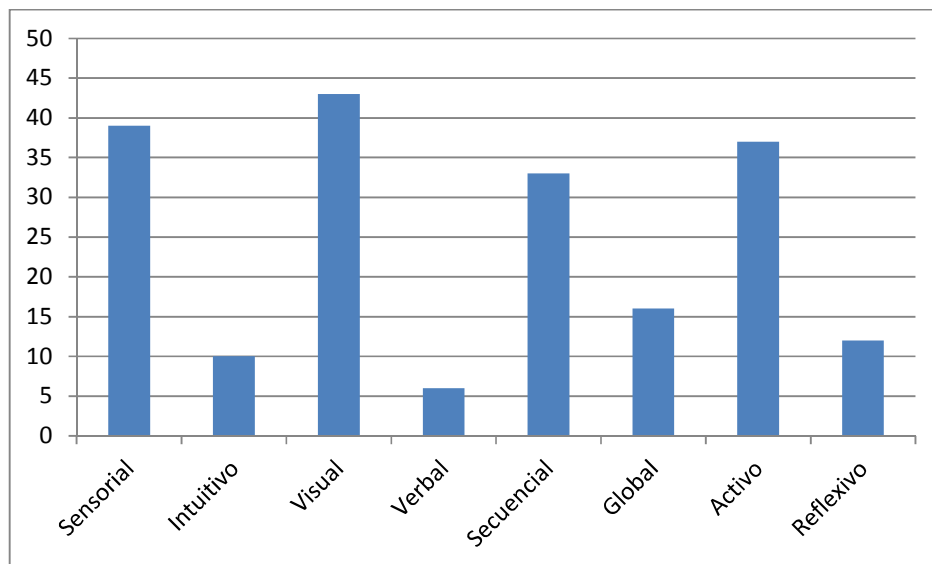


Ilustración 26. Estilos de aprendizaje para el grupo colaborativo

7.1.4 Análisis de Estilos según Preferencia

Cada estilo de aprendizaje presenta tres preferencias: moderada, equilibrada y fuerte. Una preferencia moderada por una dimensión significa que el alumno aprenderá más fácilmente en un medio de enseñanza que favorezca esa dimensión. Una preferencia fuerte por una dimensión significa que el alumno tendrá dificultad para aprender en un medio que no proporcione el entorno para dicha preferencia. Una preferencia equilibrada por una dimensión significa que el alumno podrá aprender con ambas dimensiones.

Tabla 57. Preferencias

Preferencia	Control	Experimental	Total
Moderado	35.22%	41.32%	38.27%
Equilibrado	51.13%	45.40%	48.27%
Fuerte	13.63%	13.26%	13.45%

La Tabla 57 muestra el resultado de las tendencias de los alumno, las preferencia más frecuentes en orden descendente son: Equilibrados, es decir preferencias neutras; le sigue, tendencia Moderada; en tercer lugar, está la tendencia Fuerte.

La Tabla 58 muestra la distribución de las tendencias de los estilos para cada grupo.

Tabla 58. Distribución de preferencias en los dos grupos

	Control	Experimental
Sensorial MODERADO	8%	11.2%
Intuitivo MODERADO	0	2.6%
Visual MODERADO	10.2%	8.7%
Verbal MODERADO	1.1%	0.5%
Secuencial MODERADO	5.7%	6.1%
Global MODERADO	1.1%	2.6%
Activo MODERADO	8%	8.7%
Reflexivo MODERADO	1.1%	1.0%
Sensorial EQUILIBRADO	8%	7.1%
Intuitivo EQUILIBRAO	4.5%	1.6%
Visual EQUILIBRADO	5.7%	5.6%
Verbal EQUILIBRADO	2.3%	2.0%
Secuencial EQUILIBRADO	10.8%	9.2%
Global EQUILIBRADO	5.1%	5.6%
Activo EQUILIBRADO	9.7%	9.2%
Reflexivo EQUILIBRADO	5.1%	5.1%
Sensorial FUERTE	4.5%	1.5%
Intuitivo FUERTE	0	1.0%
Visual FUERTE	5.7%	7.7%
Verbal FUERTE	0	0.5%
Secuencial FUERTE	1.7%	1.5%
Global FUERTE	0.6%	0
Activo FUERTE	1.1%	1.0%
Reflexivo FUERTE	0%	0%
Total	100%	100%

Como puede verse en la Tabla 58, con respecto al grupo de Control, el estilo secuencial equilibrado posee el mayor porcentaje (10.8%), mientras que para el grupo experimental es el estilo sensorial moderado con un porcentaje del 11.2%.

7.1.4 Conclusiones del Análisis de Estilos de Aprendizaje

En resumen, los alumnos de ambos grupos juntos como por separado se inclinan a un estilo de aprendizaje activo, sensorial, visual y secuencial. Con respecto a las preferencias:

- Estilos de preferencia Moderada: los dos grupos son sensoriales moderados, visuales moderados, secuenciales moderados y activos moderados.
- Estilos de preferencia Equilibrada: los alumnos de los dos grupos se inclinan más a ser sensoriales equilibrados, secuenciales equilibrados, activos y visuales equilibrados.
- Estilos de preferencia Fuerte: los alumnos de ambos grupos son secuenciales, activos, sensoriales y visuales fuertes.

Sabemos que los alumnos que aprenden mejor con cursos tradicionales hacen especial hincapié en las dimensiones del lado izquierdo de la Tabla 52, o sea, los estilos de aprendizaje reflexivo, intuitivo, verbal y secuencial. Con eso, podemos concluir que los grupos (control y colaborativo) se inclinan más a un aprendizaje no tradicional.

7.2 Análisis de Correlación entre Eficiencia Educativa y Estilos de Aprendizaje

Para averiguar si existe una relación entre el estilo de aprendizaje de los alumnos y el aumento de su nivel de conocimiento se ha optado por analizar la correlación entre ambas evaluaciones (Debdi, et al., 2014b).

Las evaluaciones de estilos de aprendizaje y de eficiencia se realizaron en abril de 2013 en la asignatura troncal “Diseño y Análisis de Algoritmos” del curso de Grado de Ingeniería del Software en dos grupos diferentes. El protocolo y las tareas presentadas están descritos en el apartado 5.2.

Presentamos los resultados de la correlación entre eficiencia educativa medida como la resta de las calificaciones obtenidas por los alumnos antes y después del tratamiento (*posttest- pretest*) y los estilos de aprendizaje en ambos grupos por separado y juntos. Para analizar la correlación, se comprobó primero la normalidad de las muestras, usando la correlación de *pearson* en el caso de normalidad de las muestras o la correlación de *spearman* en el caso contrario. La Tabla 59 muestra el número de alumnos participantes comunes en cada grupo. El análisis se realizó con el programa paquete estadístico SPSS 20.

Tabla 59. Alumnos comunes en el análisis de estilos y eficiencia

Grupos	Alumnos comunes
Grupo de Control	31
Grupo Colaborativo	37
Todos los grupos	68

7.2.1 Correlación entre Estilos de Aprendizaje y Eficiencia Educativa: Grupo Control

Presentamos los resultados del análisis de correlación entre la eficacia educativa y los estilos de aprendizaje para los alumnos del grupo de control.

Según la prueba de *Kolmogorov Smirnov*, las muestras Visual, Sensorial, Activo y la eficiencia siguen una distribución normal (obtención de sig > 0.05 de significación). A continuación presentamos la correlación de *pearson* entre estas muestras de estilos y eficiencia, la Tabla 60 muestra los resultados.

Tabla 60. Correlación de *pearson* para el grupo de control

		Activo	Sensorial	Visual
Eficiencia	Correlación de <i>pearson</i>	-.172	-.220	-.131
	Sig. (bilateral)	.354	.234	.483

Interpretando los resultados de la correlación de *pearson* mostrados en la Tabla 60, podemos afirmar que no existe ninguna correlación entre las muestras de los estilos Activo, Sensorial y Visual y la eficiencia educativa. A continuación, calculamos la correlación de *spearman* entre el resto de estilos que no siguen una distribución normal y la eficacia educativa mostrada en la Tabla 61.

Tabla 61. Correlación de *spearman*

		Reflexivo	Intuitivo	Verbal	Secuencial	Global
Eficiencia	Coefficiente de correlación	.354	-.020	.122	.160	-.202
	Sig. (bilateral)	.051	.916	.513	.391	.276

Según la prueba de *spearman* mostrada en la Tabla 61, no existe ninguna correlación entre los estilos de aprendizaje y la eficiencia para el grupo de control.

7.2.2 Correlación entre Estilos de Aprendizaje y Eficiencia Educativa: Grupo Colaborativo

Presentamos los resultados del análisis de correlación entre la eficacia educativa y los estilos de aprendizaje para el grupo colaborativo.

Según la prueba de *Kolmogorov Smirnov*, las muestras Activo, Sensorial, Visual, Secuencial y eficiencia siguen una distribución normal (obtención de sig > 0.05 de significación). A continuación, calculamos la correlación de *pearson* entre estas muestras en la Tabla 62.

Tabla 62. Correlación de *pearson* del grupo colaborativo

		Activo	Sensorial	Visual	Secuencial
Eficiencia	Correlación de <i>pearson</i>	-.179	.078	.064	-.066
	Sig. (bilateral)	.290	.644	.706	.696

Según la Tabla 62, no existe ninguna correlación entre las muestras de estilos y la eficiencia. La Tabla 63 presenta la correlación de *spearman* para el resto de muestras que no siguen una distribución normal.

Tabla 63. Correlación de *spearman* del grupo experimental

		Reflexivo	Intuitivo	Verbal	Global
Eficiencia	Coefficiente de correlación	.126	-.073	-.392*	-.026
	Sig. (bilateral)	.457	.669	.016	.881

Según la Tabla 63, existe una correlación negativa entre la eficiencia y el estilo de aprendizaje verbal para el grupo colaborativo.

7.2.1.3 Correlación entre Estilos de Aprendizaje y Eficiencia Educativa: Todos los Grupos

Presentamos los resultados del análisis de correlación entre la eficacia educativa y los estilos de aprendizaje para los dos grupos juntos (la suma de los alumnos de los dos grupos). Según la prueba de *Kolmogorov-Smirnov*, las muestras Reflexivo, Sensorial, Intuitivo, Verbal, Secuencial y Global no siguen una distribución normal (obtención de sig <0.05 de significación). La Tabla 64 presenta la correlación de *pearson* para el resto de estilos de aprendizaje que siguen una distribución normal.

Tabla 64. Correlación de *pearson* para todos los grupos

		Visual	Activo
Eficiencia	Correlación de <i>pearson</i>	-.006	-.150
	Sig. (bilateral)	.962	.224

Según la prueba de *pearson* mostrada en la Tabla 64, no existe ninguna correlación entre el estilo visual y activo, y el aumento de conocimiento. A continuación realizamos la prueba de *spearman* entre el resto de muestras que no presentan normalidad.

Tabla 65. Correlación de *spearman* para todos los grupos

		Reflexivo	Sensorial	Intuitivo	Verbal	Secuencial	Global
Eficiencia	Coefficiente de correlación	.198	-.030	-.046	-.176	.091	-.140
	Sig. (bilateral)	.105	.806	.710	.151	.462	.253

Según la Tabla 64, no existe ninguna correlación entre la eficiencia y estas muestras de estilos de aprendizaje.

7.2.2 Conclusiones del Análisis de Correlación entre Estilos y Eficiencia

Hemos presentado de forma detallada un análisis de correlación entre la eficacia educativa y los estilos de aprendizaje, realizado en abril 2013, habiéndose obtenido los siguientes resultados:

- Grupo de control. No existe ninguna correlación entre la eficiencia y los estilos de aprendizaje.
- Grupo colaborativo. Existe una correlación negativa entre la eficiencia y el estilo de aprendizaje verbal.
- Todos los grupos. No existe ninguna correlación entre la eficiencia y los estilos de aprendizaje.

En resumen, se puede concluir que los alumnos que aprenden menos corresponden a los alumnos cuyo estilo de aprendizaje verbal y que recibieron un esquema colaborativo, esta conclusión es lógica dado que los alumnos verbales son alumnos que requieren explicaciones escritas y verbales en lugar de diagramas o imágenes (estilo visual).

7.3 Análisis de Correlación entre Motivación y Estilos de Aprendizaje

Las evaluaciones de estilos de aprendizaje y motivación de alumnos se realizaron en abril de 2013 en la asignatura troncal “Diseño y Análisis de Algoritmos” del curso de Grado de Ingeniería

del Software en dos grupos diferentes. Después, se ha optado por analizar la correlación entre ambos (Debdi, et al., 2014c) para averiguar si existe una relación entre el estilo de aprendizaje de los alumnos y su aumento de motivación para aprender algoritmos voraces. El protocolo y las tareas presentadas están descritos en el apartado 5.2.

Presentamos los resultados de la correlación entre el aumento de motivación (*posttest-pretest*) calculado a partir de la resta de la motivación (de la media global como de cada dimensión) obtenida antes y después del tratamiento, y los estilos de aprendizaje.

Para analizar la correlación, se comprobó primero la normalidad de las muestras, usando la correlación de *pearson* en el caso de normalidad de las muestras o la correlación de *spearman* en el caso contrario. La Tabla 66 muestra el número de alumnos participantes en cada grupo. El análisis se realizó con el programa paquete estadístico SPSS 20.

Tabla 66. Alumnos comunes en motivación y estilos

Grupos	Alumnos comunes
Grupo de Control	38
Grupo Colaborativo	43
Total	81

7.3.1 Correlación entre Estilos de Aprendizaje y el Aumento de Motivación: Grupo de Control

Presentamos los resultados del análisis de correlación entre el aumento de motivación en sus cuatro dimensiones y los estilos de aprendizaje.

Según la prueba de *Kolmogorov-Smirnov*, las muestras Activo, Sensorial, Visual, media motivación, motivación intrínseca, regulación identificada y regulación externa siguen una distribución normal, (obtención de sig > 0.05 de significación). A continuación presentamos la correlación de *pearson* entre estas muestras, la Tabla 67 muestra los resultados obtenidos.

Tabla 67. Correlación de *pearson* para el grupo de control

		Activo	Sensorial	Visual
Motivación-Diferencia	Correlación de <i>pearson</i>	-.155	.171	.077
	Sig. (bilateral)	.353	.304	.646

Intrínseca-Diferencia	Correlación de <i>pearson</i>	-.257	-.001	.110
	Sig. (bilateral)	.119	.996	.512
Identificada-Diferencia	Correlación de <i>pearson</i>	-.096	.077	-.127
	Sig. (bilateral)	.565	.644	.449
Externa-Diferencia	Correlación de <i>pearson</i>	.059	-.133	-.011
	Sig. (bilateral)	.724	.426	.948

Según la correlación de *pearson* mostrada en la Tabla 67, no existe ninguna correlación entre las muestras analizadas. La Tabla 68 y la Tabla 69 presentan la correlación de *spearman* entre el resto de estilos que no siguen una distribución normal y el aumento de motivación en sus cuatro dimensiones.

Tabla 68. Correlación de *spearman* para el grupo de control

		Reflexivo	Intuitivo	Verbal	Secuencial	Global
Motivación-Diferencia	Coefficiente de correlación	-.157	-.193	.005	.029	-.044
	Sig. (bilateral)	.347	.245	.975	.861	.793
Intrínseca-Diferencia	Coefficiente de correlación	.007	.052	-.069	.072	-.248
	Sig. (bilateral)	.967	.758	.680	.668	.133
Identificada-Diferencia	Coefficiente de correlación	-.080	-.248	.208	-.010	.052
	Sig. (bilateral)	.631	.133	.210	.953	.756
Externa-Diferencia	Coefficiente de correlación	-.086	-.165	.037	-.120	.017
	Sig. (bilateral)	.608	.321	.825	.474	.918
Desmotivación-Diferencia	Coefficiente de correlación	.069	.188	.016	-.143	.054
	Sig. (bilateral)	.680	.257	.924	.392	.749

Según La Tabla 68 y la Tabla 69, no existe ninguna correlación entre el aumento de motivación y los estilos de aprendizaje para el grupo de control.

Tabla 69. Correlación de *spearman*

		Activo	Sensorial	Visual
Desmotivación-Diferencia	Coefficiente de correlación	.087	-.253	-.048
	Sig. (bilateral)	.601	.125	.774

7.3.2 Correlación entre Estilos de Aprendizaje y Aumento de Motivación: Grupo Colaborativo

Presentamos los resultados del análisis de correlación entre el aumento de motivación en sus cuatro dimensiones y los estilos de aprendizaje de los alumnos del grupo colaborativo. Según la prueba de *Kolmogorov-Smirnov*, las muestras Reflexivo, Intuitivo, Verbal, Secuencial, Global, dimensión desmotivación y motivación intrínseca no siguen una distribución normal (obtención de sig <0.05 de significación). A continuación, realizamos la correlación de *pearson* mostrada en la Tabla 70 entre el resto de estilos de aprendizaje que siguen una distribución normal y el aumento de motivación en sus cuatro dimensiones.

Tabla 70. Correlación de *pearson* del grupo Colaborativo

		Activo	Sensorial	Visual
Motivación-Diferencia	Correlación de <i>pearson</i>	-.197	.159	.106
	Sig. (bilateral)	.205	.308	.498
Identificada-Diferencia	Correlación de <i>pearson</i>	-.048	.208	.045
	Sig. (bilateral)	.758	.180	.774
Externa-Diferencia	Correlación de <i>pearson</i>	-.045	-.037	-.109
	Sig. (bilateral)	.773	.814	.487

No podemos observar ninguna correlación según la prueba de *pearson* mostrada en la Tabla 70. A continuación realizamos la prueba de *spearman* mostrada en la Tabla 71 para el resto de muestras de motivación que no siguen una distribución normal y los estilos de aprendizaje.

Tabla 71. Correlación de *spearman* del grupo Colaborativo

		Reflexivo	Intuitivo	Verbal	Secuencial	Global
Motivación-Diferencia	Coefficiente de correlación	.094	-.168	-.118	.033	-.094
	Sig. (bilateral)	.550	.283	.450	.832	.548
Intrínseca-Diferencia	Coefficiente de correlación	-.259	-.321*	.019	-.082	-.281
	Sig. (bilateral)	.094	.036	.905	.603	.068
Identificada-Diferencia	Coefficiente de correlación	.042	-.268	.156	-.002	-.062
	Sig. (bilateral)	.791	.082	.318	.990	.692
Externa-Diferencia	Coefficiente de correlación	.166	.092	-.069	.086	-.171
	Sig. (bilateral)	.288	.559	.662	.582	.273

Desmotivación-Diferencia	Coefficiente de correlación	.116	.150	-.162	.163	-.077
	Sig. (bilateral)	.459	.336	.298	.295	.625

Según la Tabla 71, existe una correlación negativa entre el aumento de motivación intrínseca y el estilo intuitivo para el grupo colaborativo. Y según la Tabla 72 no existe ninguna correlación.

Tabla 72. Correlación de *spearman*

		Activo	Sensorial	Visual
Intrínseca-Diferencia	Coefficiente de correlación	.115	.218	.212
	Sig. (bilateral)	.463	.160	.172
Desmotivación-Diferencia	Coefficiente de correlación	-.030	-.155	-.268
	Sig. (bilateral)	.847	.320	.082

7.3.3 Correlación entre Estilos de Aprendizaje y Motivación: Todos los grupos

Presentamos los resultados del análisis de correlación entre el aumento de motivación de los alumnos y los estilos de aprendizaje para los dos grupos juntos. Según la Prueba de *Kolmogorov-Smirnov*, las muestras dimensión regulación externa y estilo visual siguen una distribución normal (obtención de sig > 0.05 de significación).

Tabla 73. Correlación de *pearson* de todos los grupos

		Externa-Diferencia
Visual	Correlación de <i>pearson</i>	-.093
	Sig. (bilateral)	.409

Según los resultados mostrados en la Tabla 73, no existe una correlación entre el estilo visual y la dimensión regulación externa. La Tabla 74 muestra la correlación de *spearman* entre el aumento de motivación en sus cuatro dimensiones y los estilos de aprendizaje que no siguen una distribución normal.

Tabla 74. Correlación de *spearman* de todos los grupos

		Motivación -Diferencia	Intrínseca -Diferencia	Identificadadiferencia	Externa-Diferencia	Desmotivación-Diferencia
Activo	Coefficiente de correlación	-.174	-.071	-.120	-.005	.034

	Sig. (bilateral)	.121	.527	.287	.966	.762
Reflexivo	Coefficiente de correlación	.021	-.046	.031	.059	.081
	Sig. (bilateral)	.853	.683	.781	.598	.473
Sensorial	Coefficiente de correlación	.170	.145	.151	-.091	-.201
	Sig. (bilateral)	.130	.197	.179	.421	.072
Intuitivo	Coefficiente de correlación	-.189	-.140	-.267*	-.003	.169
	Sig. (bilateral)	.091	.213	.016	.981	.132
Visual	Coefficiente de correlación	.165	.192	-.015	--	-.152
	Sig. (bilateral)	.140	.086	.896	--	.175
Verbal	Coefficiente de correlación	-.074	-.137	.116	.043	-.019
	Sig. (bilateral)	.509	.222	.301	.705	.864
Secuencial	Coefficiente de correlación	.068	.063	.031	-.050	.012
	Sig. (bilateral)	.545	.578	.786	.657	.917
Global	Coefficiente de correlación	-.093	-.253*	-.026	-.070	-.005
	Sig. (bilateral)	.411	.023	.816	.537	.961

Según la Tabla 74, existe una correlación negativa entre el estilo de aprendizaje global y el aumento de motivación intrínseca y otra correlación negativa entre el estilo de aprendizaje intuitivo y la regulación identificada.

7.3.3 Conclusiones del Análisis de Correlación entre Estilos y Motivación

Hemos presentado de forma detallada un análisis de correlación entre la motivación de y los estilos de aprendizaje realizado en abril 2013, habiéndose obtenido los siguientes resultados.

- Grupo de Control. No existe ninguna correlación
- Grupo Colaborativo. Existe una correlación negativa entre la motivación intrínseca y el estilo intuitivo.

- Todos los grupos. Existe una correlación negativa entre el estilo de aprendizaje global y el aumento de la motivación intrínseca y una correlación negativa entre el estilo de aprendizaje intuitivo y el aumento de la regulación identificada.

En resumen, se puede concluir que los alumnos que se motivan menos corresponden a los alumnos cuyo estilo es intuitivo y que reciben un esquema colaborativo, esta conclusión no carece de significado dado que los alumnos intuitivos prefieren principios, teorías al contrario que el estilo sensorial que aprenden a través de cosas concretas, hechos y procedimiento. Por otra parte, los alumnos que se motivan menos sea cual sea el esquema que reciben corresponden a los alumnos cuyo estilo es global y intuitivo.

7.4 Conclusiones Generales

En resumen, los resultados del análisis de estilos de aprendizaje son:

Los alumnos de ambos grupos juntos como por separado se inclinan a un estilo de aprendizaje activo, sensorial, visual y secuencial. De esto, se puede concluir que los grupos analizados les favorece recibir un esquema no tradicional.

1. Activo: prefieren aprender ensayando y trabajando con otros. Se les debe proporcionar área de discusión y adivinar varias preguntas.
2. Sensorial: les gusta resolver problemas siguiendo procedimientos muy bien establecidos. Se le debe dar un ejemplo primero.
3. Visual: prefieren representaciones visuales para la obtención de información.
4. Secuencial: aprenden en pequeños pasos incrementales y bien relacionados lógicamente. Se les debe presentar el material paso a paso.

Los resultados del análisis de correlación de estilos con el aumento de motivación, y el aumento de conocimiento son los siguientes:

1. Los alumnos que se motivan menos corresponden a los alumnos cuyo estilo es intuitivo y que reciben un esquema colaborativo, esta conclusión no carece de significado dado que los alumnos intuitivos prefieren principios, teorías al contrario que el estilo sensorial que

aprenden a través de cosas concretas, hechos y procedimiento. Por otra parte, los alumnos que se motivan menos sea cual sea el esquema que reciben (juntando los dos grupos) corresponden a los alumnos cuyo estilo es global y intuitivo.

2. Los alumnos que aprenden menos corresponden a los alumnos cuyo estilo de aprendizaje es verbal y que recibieron un esquema colaborativo, esta conclusión es lógica dado que los alumnos verbales son alumnos que requieren explicaciones escritas y verbales en lugar de diagramas o imágenes (estilo visual)

De estos resultados, podemos afirmar que aunque los alumnos de ambos grupos se inclinan a ser secuenciales, globales, activos y visuales, es decir, les puede beneficiar mejor recibir un esquema no tradicional (en nuestro caso el colaborativo) no se motivan y /o aprenden MENOS por recibir clases tradicionales. Sin embargo los alumnos cuyo estilo es intuitivo y verbal (se les beneficia más recibir un esquema tradicional) y que recibieron un esquema colaborativo sí se ven perjudicados, es decir la motivación de los alumnos cuyo estilo es intuitivo se disminuye y el nivel de conocimiento de los alumnos cuyo estilo es verbal también se disminuye, al recibir un esquema colaborativo.

Capítulo 8: Conclusiones

En este capítulo, se presentan las conclusiones de este trabajo de tesis doctoral, resumiendo los puntos fundamentales de la misma y las deducciones finales del trabajo realizado llegando a obtener un nuevo método didáctico soportado por ordenador y basado en la experimentación y la colaboración. Y finalmente se recogen los trabajos futuros.

8.1 Resumen de los Resultados

El punto de partido ha sido el método experimental y las herramientas interactivas TuMiST y GreedEx para el aprendizaje de los algoritmos voraces que lo soportan. TuMiST nos permitió familiarizarnos con las características de estas herramientas, aunque GreedEx ha sido el centro de nuestra atención y desarrollo.

La principal aportación del trabajo ha sido la concepción de un nuevo método educativo que da apoyo al aprendizaje activo de los algoritmos voraces, basado en principios pedagógicos y refinados mediante su evaluación sistemática. Destacamos algunas características:

- En primer lugar, GreedEx se apoya en un método experimental para el aprendizaje activo de los algoritmos voraces basado en objetivos pedagógicos de alto nivel. La experiencia adquirida ha permitido profundizar en su uso eficaz en asignaturas de algoritmos, incluso con otras técnicas de diseño distintas de la inicialmente prevista, la técnica voraz.
- En segundo lugar, el sistema GreedEx ha sido refinado en un proceso de diseño centrado en el usuario, mejorando el sistema e incorporando nuevas funciones a través de las evaluaciones de usabilidad basados en cuestionario y registro de sesiones de los alumnos (logs) para garantizar su facilidad de uso, su adecuación a los objetivos educativos previstos y eliminar cualquier dificultad ajena a la experiencia de aprendizaje. En cuanto al método experimental, su uso docente nos ha permitido familiarizarnos con distintos casos y profundizar en su utilización. Además GreedEx ha sido ampliado con nuevos problemas de optimización.

- En tercer lugar, se extendió la herramienta GreedEx para soportar el enfoque colaborativo con el fin de obtener los beneficios del trabajo en equipo y en consecuencia mejorar la calidad del aprendizaje de los algoritmos voraces. La extensión colaborativa GreedExCol soporta la argumentación colaborativa entre un grupo de alumnos para identificar las funciones de selección óptimas. Esta extensión ayuda al alumno a experimentar con diversas funciones de selección por cada problema soportado entre un grupo de alumnos. Por un lado, el alumno debería apreciar el efecto de las diversas funciones de selección y ser capaz de realizar una elección razonada entre ellas y por otro lado, intervenir en un debate con el resto de los miembros del grupo para decidir sobre la función de selección óptima.
- En cuarto lugar, nuestro nuevo método educativo ha sido evaluado con respecto a la eficiencia educativa y la motivación de los alumnos, obteniendo mejorías estadísticamente significativas en ambos aspectos. Para garantizar que el sistema no es un impedimento sino, al contrario, un instrumento potenciador se midió el nivel de conocimiento (eficiencia educativa), antes y después de aplicar este enfoque, haciendo una comparación entre los alumnos que recibieron el esquema tradicional y el nuevo enfoque. Asimismo, se hizo una evaluación de motivación para medir el grado de motivación en las mismas circunstancias. Obteniendo resultados positivos en ambas evaluaciones, tanto en la motivación como eficiencia educativa de nuestro método educativo.
- En quinto lugar, nuestro enfoque educativo da la posibilidad de establecer o reforzar los conceptos relacionados con el método científico: hipótesis, planificación del experimento, refutación, contra-ejemplo, evidencia experimental y prueba matemática. Por otro lado, en las clases de los algoritmos voraces, el profesor debe alcanzar los objetivos de aprendizaje para cada técnica de diseño, aunque estos objetivos de aprendizaje no son fáciles de lograr con la organización de los materiales en algoritmos voraces que se puede encontrar en la mayoría de los libros de texto. Por eso la visualización de algunos problemas de la herramienta ha sido de gran ayuda para apoyar al aprendizaje de los algoritmos voraces mediante su incorporación en las transparencias de la asignatura. Además, el profesor puede utilizar GreedEx/GreedExCol para ilustrar o experimentar con muchos temas de algoritmos voraces.
- En sexto lugar, como última fase en este trabajo, se ha realizado un análisis de estilos de aprendizaje en alumnos de informática y otro análisis de correlación entre estilos de aprendizaje bien con eficiencia, bien con motivación, cuyos resultados fueron muy interesantes. De los

resultados, podemos indicar los estilos de aprendizaje de la mayoría de los alumnos de informática analizados (activo, sensorial, visual y secuencial) frente al (reflexivo, intuitivo, verbal y global), llegando a la conclusión de que nuestros alumnos de Informática les puede favorecer más recibir un esquema no tradicional para que aprenden con mayor efectividad.

- Por último, el análisis de correlación nos ha podido contestar a nuestra pregunta planteada al principio del estudio sobre si los alumnos que aprenden más o se motivan más corresponden a algún estilo de aprendizaje en concreto. Los resultados han sido que la motivación de los alumnos cuyo estilo es intuitivo se disminuye y el nivel de conocimiento de los alumnos cuyo estilo es verbal también se disminuye, al recibir un esquema colaborativo. Nuestra conclusión ha sido que si aplicamos estrategias de enseñanza de tipo visual a los alumnos verbales puede que no tengan los mejores resultados académicos. Y si aplicamos estrategias de enseñanza de tipo sensorial a los alumnos intuitivos puede que tengan menos motivación.

Cabe mencionar la innovación de nuestras aportaciones dado que no existen hasta el momento métodos que combinan el enfoque experimental y colaborativo y ser evaluados con respecto a su usabilidad, eficiencia y motivación. De esta forma, culmina este trabajo de tesis que ha aportado innovadoras contribuciones para la docencia de algoritmos voraces. Ofreciendo un método didáctico único que integra dos taxonomías pedagógicas y evaluado positivamente para terminar dándole la validez y eficiencia. Finalmente, la difusión de GreedEx/GreedExCol ha sido un proceso continuo a lo largo de esta tesis doctoral a través de comunicaciones y artículos publicados en revistas y congresos. En la página Web del departamento en el siguiente enlace <http://www.lite.etsii.urjc.es/GreedEx/> se dispone de enlaces a publicaciones de investigación e informes técnicos sobre la aplicación como la propia herramienta.

8.2 Trabajos futuros

Como trabajos futuros, podemos destacar dos líneas de investigación posibles:

1. En la dimensión colaborativa: ampliar GreedExCol para mejor soporte colaborativo añadiendo algunas herramientas de comunicación asíncrona y síncrona. En la comunicación síncrona, cuando los alumnos se están interactuando en tiempo real, por ejemplo, añadir un chat dado que ha sido una de las peticiones de los propios alumnos recogidos en las opiniones de la evaluación de usabilidad. En la comunicación asíncrona, cuando los alumnos están

interactuando en tiempos distintos, por ejemplo, creando un foro para la discusión entre alumnos o alguna herramienta de votación. De tal manera que el proceso de trabajo corriente para una sesión de prácticas usando el sistema GreedExCol quedaría de la siguiente forma: en la primera fase, los alumnos experimentan con la herramienta de forma individual. En la segunda fase, los alumnos podían experimentar de forma grupal, compartiendo el resultado de sus experimentos, enviando y recibiendo propuestas de sus compañeros y comunicándose con el resto del grupo mediante el chat. En la tercera fase, podía haber una discusión entre todos los grupos mediante el uso del foro, expresando sus opiniones sobre las funciones de selección óptimas, incluso el profesor puede plantear una pregunta a los alumnos o cuestionar la optimalidad o la no optimalidad de una función de selección para que los alumnos discuten y debaten sobre ella. En este sentido, los alumnos expresaran y justificaran sus respuestas, y el resto de alumnos, que no está en el momento del debate, podía tener acceso a ello cuando querrán, no necesariamente en tiempo real.

2. En la dimensión de informática: la limitación de GreedEx/GreedExCol es su poca generalidad para cubrir los problemas de optimización. Como trabajos futuro, se puede plantear la ampliación de GreedExCol para adaptarse a otras clases de algoritmos o crear una aplicación para generalizar los problemas de optimización, el cual se ha desarrollado una primera versión de esta herramienta, nombrada “OptimEx” (Velázquez-Iturbide, et al., 2013), que es un sistema más general para la experimentación con algoritmos de optimización que compara los resultados de diferentes algoritmos del mismo problema de optimización. En este sentido, como primera etapa, está la mejora general de de la herramienta y la creación de una versión más estable. Otra mejora es la portación del sistema OptimEx a la Web y su ampliación a nuevas funcionalidades, de modo que pueda ser utilizado por los alumnos para la autoevaluación.

Anexo 1: Enunciados de las Prácticas

En esta sección se incluyen los enunciados y los modelos del informe de las prácticas de la evaluación de GreedEx y GreedExCol.

Ingeniería Informática

Asignatura Diseño y Análisis de Algoritmos

Curso xxxx/xxxx

Sesión de introducción a GreedEx

Objetivo

El objetivo de la sesión es que el alumno se familiarice con el sistema GreedEx y experimente con la optimalidad de distintas funciones de selección para problemas sencillos.

Carácter

La sesión es voluntaria. Puede realizarse individualmente o en parejas.

Prerrequisitos

El alumno debe tener nociones básicas de algoritmos voraces, incluyendo sus elementos y el esquema de código asociado.

Realización:

GreedEx es un ayudante interactivo para la experimentación y aprendizaje de los algoritmos voraces. La sesión consta de varias fases a realizar secuencialmente, que se describen a continuación.

1. **Demostración del profesor – problema de maximizar el número de objetos.** El profesor realizará una demostración del funcionamiento de GreedEx en la que mostrará, mediante este sencillo problema, su aspecto y funciones principales.
2. **Familiarización del alumno – problema de maximizar el peso.** El alumno seleccionará el problema de maximizar el peso. Tras su experimentación, se hará una puesta en común sobre los

resultados obtenidos y de esta forma comprender claramente el objetivo y funcionamiento de GreedEx.

3. **Familiarización del alumno – problema de la mochila.** Análogo.

Ingeniería Informática

Asignatura Diseño y Análisis de Algoritmos

Curso xxxx/xxxx

Práctica nº x

Objetivo

El objetivo de la práctica es que el alumno analice el uso de varias estrategias voraces para resolver de forma óptima el problema de la selección de actividades.

Carácter

La sesión es voluntaria. Puede realizarse individualmente o en parejas, salvo el cuestionario que se realizará individualmente.

Prerrequisitos

El alumno debe tener nociones básicas de algoritmos voraces, incluyendo sus elementos y el esquema de código asociado.

Enunciado

Sea un conjunto A de n actividades $\{a_0, a_1, \dots, a_{n-1}\}$ que necesitan utilizar un recurso común, p.ej. una sala de reuniones. El recurso sólo puede ser usado por una actividad en cada momento. Cada actividad tiene un instante de comienzo c_i y un instante de finalización f_i , donde $0 \leq c_i < f_i < \infty$. Si se selecciona la actividad a_i , se desarrolla en el intervalo semiabierto de tiempo $[c_i, f_i)$. Las actividades a_i y a_j son compatibles si sus intervalos $[c_i, f_i)$ y $[c_j, f_j)$ no se solapan, es decir, si $c_i \geq f_j$ o $c_j \geq f_i$.

El *problema de selección de actividades* consiste en determinar un subconjunto de actividades compatibles cuya cardinalidad sea máxima.

Por ejemplo, sea el siguiente conjunto de actividades:

Un subconjunto S de actividades compatibles es $\{a_2, a_4, a_{10}\}$. Sin embargo, no es un subconjunto de cardinalidad máxima, como lo son $\{a_0, a_3, a_7, a_{10}\}$ y $\{a_9, a_3, a_4, a_{10}\}$. Por ejemplo, sea el siguiente conjunto de actividades:

i	0	1	2	3	4	5	6	7	8	9	10
c_i	1	2	0	5	8	5	6	8	3	2	12
f_i	4	13	6	7	12	9	10	11	8	5	14

Se pide encontrar funciones de selección voraces óptimas para este problema. La sesión de laboratorio seguirá el siguiente orden:

1. Utilización del “asistente interactivo” GreedEx para determinar una o varias funciones de selección óptimas.
2. Complimentación y entrega electrónica a Ángel Velázquez (angel.velazquez@urjc.es) de un breve informe siguiendo el modelo disponible en el sitio *web* de la asignatura.
3. Complimentación en papel y entrega de un cuestionario de opinión sobre GreedEx.



Ingeniería Informática

Asignatura Diseño y Análisis de Algoritmos

Curso xxxx/xxxx

Memoria de la práctica nº x

Alumnos:

Funciones de selección⁶ óptimas

Proponga las funciones de selección (si existe alguna) que considere óptimas para resolver el problema de la selección de actividades.

Función de selección 1: XXXXX

Función de selección 2: XXXXX

...

Justificación de la función de selección XXXXX (repítase por cada función de selección propuesta)

Justificación razonada: Se explica en términos coloquiales porqué la función de selección es óptima. Obsérvese que se espera un razonamiento sobre el resultado (óptimo) de aplicar la función de selección, no una explicación de cómo funciona o se ejecuta.

Evidencia experimental obtenida con GreedEx (resumen): Resumen de los datos de entrada probados con GreedEx y que proporcionan evidencia experimental de la optimidad de la función de selección:

Tabla de resumen o tabla abreviada:

Evidencia experimental obtenida con GreedEx (ejemplos detallados): Se incluye la ejecución detallada del algoritmo con diversas funciones de selección y varios datos de entrada, de forma que sean ilustrativos del comportamiento óptimo de las propuestas y del comportamiento no óptimo de las descartadas.

Tabla de datos de entrada:

Tabla de resultados con todas las funciones de selección ejecutadas:

Ejecución de cada función de selección: (opcional, repítase por cada una)

Incluirá dos visualizaciones de los datos de entrada, una del estado inicial y otra del estado final tras aplicar la función de selección

Anexo 2: Cuestionario de Usabilidad

Nombre y apellidos (opcional): _____

Por favor marca, en cada pregunta, un valor de la escala mostrada en la siguiente tabla. Según la clase de pregunta, su significado se referirá a opinión o calidad:

Valor	Opinión	Calidad
1	Nada de acuerdo	Muy mala
2	Poco de acuerdo	Mala
3	Sin opinión	Regular
4	Algo de acuerdo	Buena
5	Totalmente de acuerdo	Muy buena

Facilidad de uso de GreedEx:

___ Si te parece que GreedEx es *fácil de usar*

Las partes que te parecen más difíciles de usar (si las hay) son:

Utilidad de GreedEx:

Si te parece que GreedEx *te ha ayudado*, en el problema de la selección de actividades, a:

Analizar el efecto de las distintas estrategias voraces

Identificar una estrategia óptima

Calidad de GreedEx:

Si te parece alta *la calidad en general* de GreedEx para analizar el efecto de distintas estrategia voraces

Si te parece alta *la calidad de varios aspectos* de GreedEx:

Estructura del menú principal

Iconos

Panel de visualización

Pestaña de problema

Pestaña de algoritmo

Pestaña de tabla de datos de entrada

Pestaña de tabla de resultados

Pestaña de tabla de resumen global

Introducción o generación de datos de entrada

Ejecución/animación del algoritmo

Selección de estrategias

Exportación

Preguntas generales:

___ Si en conjunto *te ha gustado* GreedEx

Responde a las siguientes preguntas en formato libre:

1. Di qué características te parece que podrían ser útiles pero GreedEx carece de ellas:

2. Di qué características de GreedEx te parecen tan poco útiles que las suprimirías:

3. Describe los aspectos positivos que encuentras en GreedEx (sobre todo si no se han mencionado antes):

4. Describe los aspectos negativos que encuentras en GreedEx (sobre todo si no se han mencionado antes)

Anexo 3: Cuestionario de Motivación 2012

Este cuestionario es para evaluar el grado en que te ha motivado la actividad o tarea que has hecho en clase usando la herramienta GreedEx. La nota NO depende en ningún caso de tus respuestas en este cuestionario. Este cuestionario se trata de forma totalmente anónima. El cuestionario consiste en contestar a la pregunta: ¿Por qué crees que debías realizar la actividad de usar GreedEx para estudiar los algoritmos voraces?

Te proporcionamos 14 afirmaciones como respuesta a esta pregunta y tú debes valorar el grado en el que estás de acuerdo con cada una de ellas con la escala que va del 1 al 7:

1 (no se corresponde en absoluto con lo que pienso)

...

4 (se corresponde al 50% con lo que pienso)

...

7 (se corresponde exactamente con lo que pienso)

Por favor, es imprescindible que contestes a todas la afirmaciones y sé sincero/a en tus valoraciones, de lo contrario este cuestionario no serviría para nada. Gracias por tu colaboración.

¿Por qué crees que debías realizar la actividad de usar GreedEx para estudiar los algoritmos voraces? ____

1. Porque creo que esta actividad es interesante ____
2. Por mi propio bien ____
3. Porque se supone que debo hacerlo ____
4. Puede que haya buenas razones para realizar esta actividad, pero yo no veo ninguna ____
5. Porque disfruto con esta actividad ____
6. Porque creo que esta actividad es buena para mí ____
7. Porque es algo que tengo que hacer ____
8. Realizo esta actividad, pero no estoy seguro de si vale la pena ____
9. Porque esta actividad es divertida ____
10. No lo sé, no veo qué me aporta esta actividad ____

11. Porque me siento bien realizando esta actividad ____
12. Porque creo que esta actividad es importante para mí ____
13. Porque creo que tengo que hacerlo ____

Hago esta actividad, pero no estoy seguro de que sea conveniente continuar con ella ____

Anexo 4: Cuestionario de Motivación 2013

Este cuestionario es para evaluar el grado en que te ha motivado la actividad o tarea que has hecho en clase o que vas a realizar. El cuestionario consiste en contestar a la pregunta.

Te proporcionamos 14 afirmaciones como respuesta a esta pregunta y tú debes valorar el grado en el que estás de acuerdo con cada una de ellas con la escala que va del 1 al 7:

1 (no se corresponde en absoluto con lo que pienso)

...

4 (se corresponde al 50% con lo que pienso)

...

7 (se corresponde exactamente con lo que pienso)

Por favor, es imprescindible que contestes a todas las afirmaciones y sé sincero/a en tus valoraciones, de lo contrario este cuestionario no serviría para nada. Gracias por tu colaboración.

Pregunta *pretest*: Teniendo en cuenta lo que ha explicado el profesor que se va a hacer en clase ¿Por qué crees que deberás realizar la actividad que vais a hacer en clase (experimentar con algoritmos voraces)?

Pregunta *posttest*: ¿Por qué crees que deberás realizar la actividad que habéis hecho en las clases de la práctica de voraces en el laboratorio?

1. Porque creo que esta actividad es interesante ____
2. Por mi propio bien ____
3. Porque se supone que debo hacerlo ____
4. Puede que haya buenas razones para realizar esta actividad, pero yo no veo ninguna ____
5. Porque disfruto con esta actividad ____
6. Porque creo que esta actividad es buena para mí ____
7. Porque es algo que tengo que hacer ____
8. Realizo esta actividad, pero no estoy seguro de si vale la pena ____
9. Porque esta actividad es divertida ____

10. No lo sé, no veo qué me aporta esta actividad ____
11. Porque me siento bien realizando esta actividad ____
12. Porque creo que esta actividad es importante para mí ____
13. Porque creo que tengo que hacerlo ____
14. Hago esta actividad, pero no estoy seguro de que sea conveniente continuar con ella ____

Anexo 5: Prueba de Conocimiento

Nombre y apellidos:

Titulación/grupo:

Número de Expediente:

El *test* es voluntario y permite aumentar tu nota de prácticas si has rellenado completamente todos sus apartados.

IMPORTANTE: No basta con que contestes sí o no. Debes completar cada respuesta o, al menos, decir claramente que no la encuentras.

1. Los problemas de optimización buscan aumentar beneficios o reducir costes mediante la toma de unas decisiones entre todas las posibles. Por ejemplo, se puede querer encontrar el camino más corto entre dos ciudades. Dado un problema de optimización, ¿sólo hay una solución óptima o puede haber varias en general?

¿Por qué?

2. Y sobre el método o algoritmo que pueda resolver un problema de optimización, ¿sólo hay un algoritmo posible o puede haber varios en general?

¿Por qué?

3. Di si conoces el nombre de los algoritmos siguientes:

○ Algoritmo de Dijkstra:

Sí No

En caso afirmativo, indica qué problema resuelve:

○ Algoritmo de Prim

Sí No

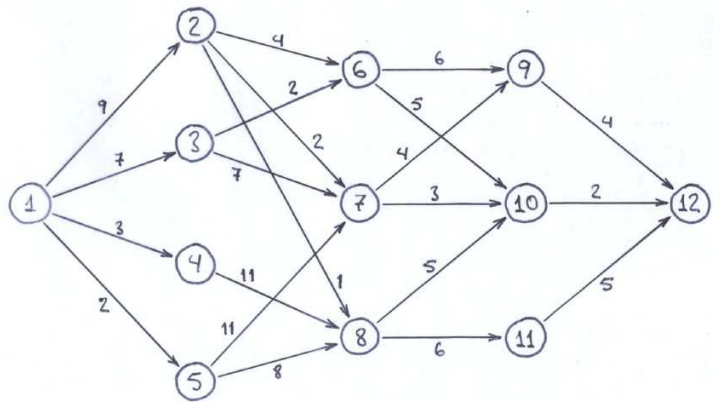
En caso afirmativo, indica qué problema resuelve:

○ Algoritmo de Kruskal

Sí No

En caso afirmativo, indica qué problema resuelve:

4. Sea el siguiente grafo cuyos nodos se encuentran dispuestos en cinco grupos verticales:



El problema consiste en encontrar un camino de longitud mínima entre el nodo origen (nodo 1) y el nodo destino (nodo 12). Por la forma del grafo, dicho camino óptimo constará de cuatro “fases”, siendo cada fase un arco que une un nodo de un grupo a otro nodo del grupo situado inmediatamente a su derecha. En general, para un grafo de este estilo, ¿crees que debe haber un único camino óptimo o que podría haber varios?

¿Por qué?

Dicho camino óptimo, ¿crees que puede encontrarse en una sola iteración de izquierda a derecha, tomando un arco de cada fase, o que debe analizarse todo el grafo de alguna forma más exhaustiva?

5. Tenemos n trabajos, donde cada trabajo i tiene una fecha tope de realización $f_i > 0$ y un beneficio $b_i > 0$. El beneficio b_i del trabajo i se gana si y sólo si se empieza a realizar antes o coincidiendo con su fecha tope f_i . Cada trabajo necesita una unidad de tiempo para realizarse; además, en un instante de tiempo sólo se puede ejecutar una tarea. Queremos determinar los trabajos a realizar (y en qué orden) para que el beneficio total sea máximo.

Por ejemplo, sean 4 actividades con las fechas tope y los beneficios contenidos en la siguiente tabla. Tres soluciones posibles serían $\{2,1\}$, $\{1,3\}$, $\{3,1\}$; sin embargo, la solución de mayor beneficio sería $\{4,1\}$.

Núm. trabajo	1	2	3	4
Beneficios	100	10	15	7
Fechas tope	2	1	2	1

¿Cuál o cuáles de las siguientes estrategias crees que encuentra soluciones óptimas (índice es el número de trabajo)?

- Selección por orden de menor a mayor índice
- Selección por orden de mayor a menor índice
- Selección por orden de menor a mayor beneficio
- Selección por orden de mayor a menor beneficio
- Selección por orden de menor a mayor fecha tope
- Selección por orden de mayor a menor fecha tope

Respalda tu respuesta con algún ejemplo o razonamiento.

6. Sean dos vectores a y b de n números naturales cada uno, de los cuales a está ordenado. Se desea obtener una reordenación de la secuencia b tal que maximice la suma de los elementos de a que no superen al elemento correspondiente de b , es decir:

$$\text{maximizar } \sum a_i \mid a_i \leq b_i$$

Por ejemplo, dados:

$$a = \{2, 6, 7, 10, 22\}$$

$$b = \{15, 1, 7, 5, 2\}$$

una reordenación óptima de b es:

$$b = \{5, 2, 7, 15, 1\}$$

con un beneficio $2+7+10 = 19$. Sin embargo, no es óptima la reordenación:

$$b = \{1, 2, 5, 7, 15\}$$

con un beneficio 0.

Un algoritmo sencillo consistiría en coger los elementos de a en orden decreciente. A cada elemento de a se le hace corresponder el mayor de los elementos que quedan de b , si éste es mayor, o el menor de los elementos que quedan de b , en caso contrario.

En el ejemplo anterior, en el primer paso no se hace corresponder al valor 22 de a con el mayor valor de b (15) porque éste es menor, sino con el valor menor de b (1). Sin embargo, en el

segundo paso, se hace corresponder el siguiente valor de a (10) con el mayor valor de b (15) porque éste es mayor.

¿Crees que el algoritmo es óptimo siempre?

- Sí No

Respalda tu respuesta con algún ejemplo o razonamiento.

Anexo 6: Cuestionario de Estilos de Aprendizaje

Encierre en un círculo la opción “a” o “b” para indicar su respuesta a cada pregunta. Por favor seleccione solamente una respuesta para cada pregunta. Si tanto “a” y “b” parecen aplicarse a usted, seleccione aquella que se aplique más frecuentemente.

1. Entiendo mejor algo

- a) si lo practico
- b) si pienso en ello

2. Me considero

- a) realista
- b) innovador

3. Cuando pienso acerca de lo que hice ayer, es más probable que lo haga en base

- a) una imagen
- b) palabras

4. Tengo tendencia a entender

- a) los detalles de un tema pero no ver claramente su estructura completa
- b) la estructura completa pero no ver claramente los detalles

5. Cuando estoy aprendiendo algo nuevo, me ayuda

- a) hablar de ello
- b) pensar en ello

6. Si yo fuera profesor, yo preferiría dar un curso

- a) que trate sobre hechos y situaciones reales de la vida
- b) que trate con ideas y teorías

7. Prefiero obtener información nueva de

- a) imágenes, diagramas, gráficos o mapas

b) instrucciones escritas o información verbal

8. Una vez que entiendo

a) todas las partes, entiendo el total

b) el total de algo, entiendo como encajan sus partes

9. En un grupo de estudio que trabaja con un material difícil, es más probable que

a) participe y contribuya con ideas

b) no participe y solo escuche

10. Es más fácil para mí

a) aprender hechos

b) aprender conceptos

11. En un libro con muchas imágenes y gráficas es más probable que

a) revise cuidadosamente las imágenes y las gráficas

b) me concentre en el texto escrito

12. Cuando resuelvo problemas de matemáticas

a) generalmente trabajo sobre las soluciones con un paso a la vez

b) frecuentemente sé cuáles son las soluciones, pero luego tengo dificultad para imaginarme los pasos para llegar a ellas

13. En las clases a las que he asistido

a) he llegado a saber cómo son muchos de los estudiantes

b) raramente he llegado a saber cómo son muchos estudiantes

14. Cuando leo temas que no son de ficción, prefiero

a) algo que me enseñe nuevos hechos o me diga como hacer algo

b) algo que me dé nuevas ideas en que pensar

15. Me gustan los profesores

a) que utilizan muchos esquemas en el pizarrón

b) que toman mucho tiempo para explicar

16. Cuando estoy analizando un cuento o una novela

a) pienso en los incidentes y trato de acomodarlos para configurar los temas

b) me doy cuenta de cuáles son los temas cuando termino de leer y luego tengo que regresar y encontrar los incidentes que los demuestran

17. Cuando comienzo a resolver un problema de tarea, es más probable que

a) comience a trabajar en su solución inmediatamente

b) primero trate de entender completamente el problema

18. Prefiero la idea de

a) certeza

b) teoría

19. Recuerdo mejor

a) lo que veo

b) lo que oigo

20. Es más importante para mí que un profesor

a) exponga el material en pasos secuenciales claros

b) me dé un panorama general y relacione el material con otros temas

21. Prefiero estudiar

a) en un grupo de estudio

b) solo

22. Me considero

a) cuidadoso en los detalles de mi trabajo

b) creativo en la forma en la que hago mi trabajo

23. Cuando alguien me da direcciones de nuevos lugares, prefiero

a) un mapa

b) instrucciones escritas

24. Aprendo

a) a un paso constante. Si estudio con ahínco consigo lo que deseo

b) en inicios y pausas. Me llevo a confundir y súbitamente lo entiendo

25. Prefiero primero

a) hacer algo y ver que sucede

b) pensar como voy a hacer algo

26. Cuando leo por diversión, me gustan los escritores que

a) dicen claramente lo que desean dar a entender

b) dicen las cosas en forma creativa e interesante

27. Cuando veo un esquema o bosquejo en clase, es más probable que recuerde

a) la imagen

b) lo que el profesor dijo acerca de ella

28. Cuando me enfrento a un cuerpo de información

a) me concentro en los detalles y pierdo de vista el total de la misma

b) trato de entender el todo antes de ir a los detalles

29. Recuerdo más fácilmente

a) algo que he hecho

b) algo en lo que he pensado mucho

30. Cuando tengo que hacer un trabajo, prefiero

a) dominar una forma de hacerlo

b) intentar nuevas formas de hacerlo

31. Cuando alguien me enseña datos, prefiero

a) gráficos

b) resúmenes con texto

32. Cuando escribo un trabajo, es más probable que lo haga (piense o escriba)

a) desde el principio y avance

b) en diferentes partes y luego las ordene

33. Cuando tengo que trabajar en un proyecto de grupo, primero quiero realizar

a) una “tormenta de ideas “donde cada uno contribuye con ideas

b) la “tormenta de ideas “en forma personal y luego juntarme con el grupo para comparar las ideas

34. Considero que es mejor elogio llamar a alguien

a) sensible

b) imaginativo

35. Cuando conozco gente en una fiesta, es más probable que recuerde

a) cómo es su apariencia

b) lo que dicen de sí mismos

36. Cuando estoy aprendiendo un tema, prefiero

a) mantenerme concentrado en ese tema, aprendiendo lo más que pueda de él

b) hacer conexiones entre ese tema y temas relacionados

37. Me considero

a) abierto

b) reservado

38. Prefiero cursos que dan más importancia a

a) material concreto (hechos, datos)

b) material abstracto (conceptos, teorías)

39. Para divertirme, prefiero

a) ver televisión

b) leer un libro

40. Algunos profesores inician sus clases haciendo un bosquejo de lo que enseñarán. Esos bosquejos son

a) algo útiles para mí

b) muy útiles para mí

41. La idea de hacer una tarea en grupo con una sola calificación para todos

a) me parece bien

b) no me parece bien

42. Cuando hago grandes cálculos

a) tiendo a repetir todos mis pasos y revisar cuidadosamente mi trabajo

b) me cansa hacer su revisión y tengo que esforzarme para hacerlo

43. Tiendo a recordar lugares en los que he estado

a) fácilmente y con bastante exactitud

b) con dificultad y sin mucho detalle

44. Cuando resuelvo problemas en grupo, es más probable que yo

a) piense en los pasos para la solución de los problemas

b) piense en las posibles consecuencias o aplicaciones de la solución en un amplio rango de campos

Referencias

- Abran, A., Khelifi, A., Suryan, W. & Seffah, A., 2003. Usability Meanings and Interpretations in ISO Standards. *Software Quality Journal*, 11(4), pp. 325-338.
- Albion, P. R., 1999. Heuristic evaluation of educational multimedia: From theory to practice. *16th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education*, pp. 5-8.
- Almeida, F., Blanco, V. & Moreno, L., 2003. EDApplets: Una Herramienta Web para la Enseñanza de Estructuras de datos y Técnicas Algorítmicas. *X Jornadas de Enseñanza Universitaria de la Informática, Universidad de Laguna*, pp. 409-417.
- Alonso, C. M., Gallego, D. J. & Honey, P., 1994. *Los estilos de aprendizaje: procedimientos de diagnósticos y mejora*. s.l.:Mensajero.
- Alsumait, A. & Al-Osaimi, A., 2009. Usability Heuristics Evaluation for Child E-learning Applications. *11th International Conference on Information Integration and Web-based Applications & Services*, pp. 425-430.
- Alsuwaiyel, M. H., 1999. *Algorithms Design Techniques and Analysis*, s.l.: World Scientific .
- Amrit, C., Hillegersberg, J. V. & Harmsen, F., 2007. 1st International Workshop on Tools for Managing Globally Distributed Software Development (TOMAG 2007). *The Seventh ICGSE '07 Proceedings of the International Conference on Global Software Engineering*, pp. 278-279.
- Anderson, L. y otros, 2001. *A Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. New York: Pearson.
- Anon., 2009. *The Algorithm Visualization Portal*. [En línea] Available at: www.AlgoViz.org [Último acceso: 06 11 2013].
- Anon., 2013. *Programming Contest Control System PC2*. [En línea] Available at: <http://www.ecs.csus.edu/pc2/> [Último acceso: 11 11 2013].

- Armoni, M., 2006. On the Role of Proofs in a Course on Design and Analysis of Algorithms. *ITiCSE-WGR '06 Working group reports on ITiCSE on Innovation and technology in computer science education*, pp. 39 - 42.
- Arroyo, J. G. & Artacho, M. R., 2000. *Esquemas algorítmicos enfoque metodológico y problemas resueltos*. s.l.:UNED.
- Baase, S. & Gelder, A. V., 2000. *Computer Algorithms: Introduction to Design and Analysis*. s.l.:Addison-Wesley.
- Baldwin, D., 1992. Using Scientific Experiments in Early Computer Science Laboratories. *The twenty-third SIGCSE technical symposium on Computer science education (SIGCSE '92)*, 24(1), pp. 102-106.
- Barab, S. & Duffy, T., 2000. From Practice Fields to Communities of Practice. En: *Theoretical Foundations of Learning Environments*. s.l.:Lawrence Erlbaum Associates , pp. 25-55.
- Barrows, H. & Tamblyn, R., 1980. *Problem-based learning: An approach to medical education*. New York: Springer.
- BECTA, 2005. *Learning styles : an introduction to the research literature*. [En línea] Available at: <http://dera.ioe.ac.uk/id/eprint/14118> [Último acceso: 20 5 2014].
- Benson, L. y otros, 2002. Usability and instructional Design Heuristics for E-learning Evaluation. *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pp. 1615-1621.
- Blecken, A. & Marx, W., 2010. Usability Evaluation of a Learning Management System. *Proceedings of the 43rd Hawaii International Conference on System Sciences*, pp. 1-9.
- Blumenfeld, p. y otros, 1993. Motivating project-based learning: Sustaining the doing, supporting the learning. *Educational Psychologist*, 26(3-4), pp. 369-398.
- Brassard, G., 2001. *Fundamentos de algoritmia*. s.l.:Prentice-Hall.
- Brassard, G. & Bratley, P., 1996. *Fundamentals of Algorithmics*. s.l.:Prentice-Hall.

Bridgeman, S., Goodrich, M., Kobourov, S. & Tamassia, R., 2000. PILOT: An interactive tool for learning and grading. *The 31st SIGCSE Technical Symposium on Computer Science Education (SIGCSE'00)*, pp. 139-143.

Bruner, J. S., 1976. *On knowing: Essays for the left hand*. s.l.:Atheneum.

Bruner, J. S. B., 1977. *The Process of Education*. s.l.:Harvard University Press.

Capdeferro, N. & Romero, M., 2012. Are online learners frustrated with collaborative learning experiences?. *The International Review Of Research In Open And Distance Learning*, 13(2), pp. 26-44.

Cassidy, S., 2004. Learning styles: an overview of theories, models and measures. *Educational Psychology*, 24(4), pp. 419-444.

Chang, T. J. & Jehng, J., 1998. Designing Computer Support for Collaborative Visual Learning in the Domain of Computer Programming. *Computers in Human Behavior*, 14(3), pp. 429-448.

Chang, W.-C. & Chen, K.-C., 2008. Collaborative Learning Tool Applying to C Programming Language. *ICWL '08 Proceedings of the 7th International Conference on Advances in Web Based Learning*, pp. 178 - 186.

Cheang, B., Kurnia, A., Lim, A. & Oon, W., 2003. On automated grading of programming assignments in an academic institution. *Computers & Education*, 41(2), pp. 121 - 131 .

Chen, M.-Y., Wei, J. D., Huang, J. H. & Lee, D., 2006. Design and applications of an Algorithm benchmark system in a computational problem solving environment. *11th annual conference on innovation and technology in computer science education (ITiCSE 2006)*, pp. 123-127.

Chin, L. S., Worth, D. & Greenough, C., 2007. *A survey of software testing tools for computational science*, s.l.: RAL Technical Reports, RAL-TR-2007-010.

Chinn, D., 2005. Peer Assessment in the Algorithms Course. *ITiCSE '05 Proceedings of the 10th annual SIGCSE conference on Innovation and Technology in Computer Science Education*, 37(3), p. 69-73.

Chow, A. & Law, N., 2005. Measuring Motivation in Collaborative Inquiry-Based Learning Contexts. *CSCL '05 Proceedings of the 2005 conference on Computer Support for Collaborative Learning: learning 2005: the next 10 years!*, pp. 68-75.

Ciesielski, V. & McDonald, P., 2001. Using Animation of State Space Algorithms to Overcome Student Learning Difficulties. *The 6th annual conference on Innovation and Technology in Computer Science Education (ITiCSE '01)*, pp. 97-100.

Cohen, L., Manion, L. & Morrison, K., 2007. Strategies for data collection and researching - Observation. En: 5, ed. *Research Methods in Education*. s.l.:Taylor & Francis, pp. 396-412.

Collberg, C., Kobourov, S. G., Miller, J. & Westbrook, S., 2002. AlgoVista: a tool to enhance algorithm design and understanding. *ITiCSE '02 Proceedings of the 7th annual conference on Innovation and Technology in Computer Science Education*, pp. 228-228.

Collberg, C., Kobourov, S. G. & Westbrook, S., 2004. AlgoVista: an Algorithmic Search Tool in an Educational Setting. *SIGCSE '04 Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education*, pp. 462-466.

Cormen, T. H., Leiserson, C. E., Rivest, R. L. & Stein, C., 2001. *Introduction to Algorithms*. s.l.:The MIT Press.

Costa, G., D'Ambrosio, C. & Martello, S., 2010. A Free Educational Java Framework for Graph Algorithms. *Journal of Computer Science*, 6(1), pp. 87-91.

Crescenzi, P. & Nocentini, C., 2007. Fully Integrating Algorithm Visualization into a CS2 Course: A Two-Year Experience. *Annual SIGCSE Conf. Innovation and Technology in Computer Science Education (ITiCSE 2007)*, pp. 296-300.

Cross, J. H. & Hendrix, T. D., 2007. jGRASP: an integrated development environment with visualizations for teaching Java in CS1, CS2, and beyond.. *Journal of Computing Sciences in Colleges*, Volumen 23, pp. 169-171.

Dagdilelis, V. & Satratzemi, M., 1998. DIDAGRAPH: Software for teaching graph theory algorithms. *The 3rd Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE'98)*, pp. 64-68.

D'Antonio, L., 2003 . Incorporating Bioinformatics in an Algorithms Course. *The 8th annual conference on Innovation and technology in computer science education (ITiCSE '03)*, pp. 211 - 214.

De Bello, T. C., 2006. Comparison of Eleven Major Learning Styles Models: Variables; Appropriate Populations; Validity of Instrumentation; and the Research Behind Them. *Journal of Reading, Writing, and Learning Disabilities International*, 6(3), pp. 203-222.

Debdi, O., Granada, J. D. & Velázquez-Iturbide, J. Á., 2010b. Ayudante interactivo para los algoritmos de Prim y Kruskal. *XVI Jornadas de Enseñanza Universitaria de la Informática (JENUI 2010)*, pp. 469-477.

Debdi, O., Paredes-Velasco, M. & Velázquez- Iturbide, J. Á., 2014b. *Análisis de Correlación entre Eficiencia Educativa y Estilos de Aprendizaje*, Madrid: Serie de Informes Técnicos DLSII-URJC, Número 2014-06, Departamento de Lenguajes y Sistemas Informáticos I.

Debdi, O., Paredes-Velasco, M. & Velázquez-Iturbide, J. Á., 2013a. *Un Análisis de Correlación entre Motivación y Eficiencia Educativa con GreedExCol*, Madrid: Serie de Informes Técnicos DLSII-URJC, Número 2013-06, Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.

Debdi, O., Paredes-Velasco, M. & Velázquez-Iturbide, J. Á., 2013c. *Un Análisis de Correlación entre Motivación y Usabilidad con GreedExCol*, Madrid: Serie de Informes Técnicos DLSII-URJC, Número 2013-04, Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.

Debdi, O., Paredes-Velasco, M. & Velázquez-Iturbide, J. Á., 2013e. *Una Segunda Evaluación de Motivación de GreedExCol*, Madrid: Serie de Informes Técnicos DLSII-URJC, Número 2013-05, Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.

Debdi, O., Paredes-Velasco, M. & Velázquez-Iturbide, J. Á., 2014a. *Un Análisis de Estilos de Aprendizaje en Alumnos de Informática*, Madrid: Serie de Informes Técnicos DLSII-URJC, Número 2014-05, Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.

Debdi, O., Paredes-Velasco, M. & Velázquez-Iturbide, J. Á., 2014c. *Análisis de Correlación entre Motivación y Estilos de Aprendizaje*, Madrid: Serie de Informes Técnicos DLSII-URJC, Número 2014-07. Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.

Debdi, O., Paredes-Velasco, M. & Velázquez, J. Á., 2013b. *Una Evaluación de Eficacia Educativa de GreedExCol*, Madrid: Serie de Informes Técnicos DLSI1-URJC, Número 2013-02, Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.

Debdi, O. & Velázquez-Iturbide, J., 2010c. Trabajos para la mejora de ayudantes interactivos para la docencia de algoritmos combinatorios. *Actas del IV Seminario de Investigación en Tecnologías de la Información Aplicadas a la Educación (SITIAE 2010)*, pp. 115-146.

Debdi, O. & Velázquez-Iturbide, J. Á., 2010a. Experimentación interactiva con algoritmos voraces. *XVI Jornadas de Enseñanza Universitaria de la Informática (JENUI 2010)*, pp. 485-492.

Debdi, O. & Velázquez-Iturbide, J. Á., 2011a. Evaluación de Usabilidad en Sistemas de Uso Educativo. *Actas del V Seminario de Investigación en Tecnologías de la Información Aplicadas a la Educación (SITIAE 2011)*.

Debdi, O. & Velázquez-Iturbide, J. Á., 2011b. *Una Evaluación de Usabilidad de GreedEx*, Madrid: Serie de Informes Técnicos DLSI1-URJC, Número 2011-01, Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.

Debdi, O., Velázquez-Iturbide, J. Á., Esteban-Sánchez, N. & Pérez-Carrasco, A., 2012. Observations as a Method to Evaluate a Computer-Based Approach to Learning Algorithms. *The 12th IEEE International Conference on Advanced Learning Technologies (ICALT'12)*, IEEE Computer Society, pp. 433-435.

Debdi, O., Velázquez-Iturbide, J. Á. & Paredes-Velasco, M., 2010. *Revisión Bibliográfica de Problemas Combinatorios Resolubles por Técnicas Básicas de Diseño de Algoritmos*, Madrid: Serie de Informes Técnicos DLSI1-URJC, Número 2010-03, Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.

Debdi, O., Velázquez-Iturbide, J. Á. & Paredes-Velasco, M., 2012c. *Una Evaluación de Usabilidad de GreedExCol*, Madrid: Serie de Informes Técnicos DLSI1-URJC, Número 2012-05, Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.

Debdi, O., Velázquez-Iturbide, J. Á. & Paredes-Velasco, M., 2013f. *Una Evaluación de Motivación de GreedExCol*, Madrid: Serie de Informes Técnicos DLSI1-URJC, Número 2013-03, Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.

Debdi, O. & Velázquez, J. Á., 2012b. *Una Segunda Evaluación de Usabilidad de GreedEx*, Madrid: Serie de Informes Técnicos DLSII-URJC, Número 2012-01, Departamento de Lenguajes y Sistemas Informáticos I, Universidad Rey Juan Carlos.

Deci, E. L. & Ryan, R. M., 1985. *Intrinsic motivation and selfdetermination in human behavior*. New York: Plenum.

Denning, P. y otros, 1989. Computing as a discipline. *Communications of the ACM*, 32(1), pp. 9-23.

Diehl, S., 2007. *Software Visualization: Visualizing the Structure, Behaviour, and Evolution of Software*. s.l.:Springer.

Dillenbourg, P., 1999. *Introduction: What do you mean by collaborative learning? Collaborative learning: Cognitive and Computational Approaches*. s.l.:Elsevier.

Dodds, Z., Libeskind-Hadas, R. & Bush, E., 2012. Bio1 as CS1: evaluating a crossdisciplinary CS context. *The 17th ACM annual conference on Innovation and technology in computer science education (ITiCSE '12)*, pp. 268-272.

Dodl, N., 1990. Instructional groupware: Design considerations. *32nd Annual International Conference of the Association for the Development of*, pp. 344-352.

Douce, C., Livingstone, D. & Orwell, J., 2005. Automatic Test-Based Assessment of programming: A review. *Journal on Educational Resources in Computing (JERIC)*, 5(3), p. No. 4.

Douce, C. y otros, 2005. A Technical Perspective on ASAP – Automated System for Assessment of Programming. *9th Computer Assisted Assessment (CAA) Conference, Loughborough, U.K.*.

Eales, R. T. J., Hall, T. & Bannon, L. J., 2002. The Motivation is the Message: Comparing CSCL in different settings. *CSCL '02 Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community*, pp. 310-317 .

Ellis, A. y otros, 1998. Resources, tools, and techniques for problem based learning in Computing. *Innovation and Technology in Computer Science Education. ITiCSE'98 Working Group Reports*, 26(4), pp. 41-56.

Ertmer, A. & Newby, J., 1993. Conductismo, cognitivismo y constructivismo: una comparación de los aspectos críticos desde la perspectiva del diseño de instrucción. *Performance Improvement Quarterly*, pp. 50-72.

Esteban Sánchez, N., Pizarro, C. & Velázquez-Iturbide, J. Á., 2014. Evaluation of a Didactic Method for the Active Learning of Greedy Algorithms. *IEEE Transactions on Education*, 57(2), pp. 83-91.

Felder, R. M. & Silverman, L. K., 1988. Learning and teaching styles in engineering education. *Engineering Education*, 78(7), pp. 674-681.

Felder, R. & Spurlin, J., 2005. Applications, Reliability and Validity of the Index of Learning Styles. *International Journal of Engineering*, 21(1), pp. 103-112.

Fenwick, J. B., Norris, C. & Wilkes, J., 2002. Scientific Experimentation via the Matching Game. *The 33rd SIGCSE technical symposium on Computer science education (SIGCSE '02)*, pp. 326-330.

Fincher, S. & Petre, M., 2004. Mapping the Territory. En: *Computer Science Education Research*. s.l.:s.n.

Fisker, K., McCall, D., Kölling, M. & Quig, B., 2008. Group work support for the BlueJ IDE. *ITiCSE '08 Proceedings of the 13th annual conference on Innovation and Technology in Computer Science Education*, pp. 163-168.

Gamma, E. y otros, 1994. *Patrones de diseño*. s.l.:s.n.

García-Mateos, G. & Fernández-Alemán, J. L., 2009. A course on algorithms and data structures using on-line judging. *The 14th annual ACM SIGCSE conference on Innovation and Technology in Computer Science Education (ITiCSE '09)*, 41(3), pp. 45-49.

Gardner, H., 1993. *Multiple Intelligences: Gardner's Theory*. New York: Basic Books.

Gerdt, P., Kommers, P., Looi, C. K. & Sutinen, E., 2001. Woven stories as a cognitive tool. *CT '01 Proceedings of the 4th International Conference on Cognitive Technology: Instruments of Mind*, p. 233-247.

Giménez, O., Petit, J. & Roura, S., 2012. Judge.org: An educational programming judge. *The 43th SIGCSE Technical Symposium on Computer Science Education (SIGCSE'12)*, pp. 445-450.

Ginat, D., 2001. Misleading Intuition in Algorithmic Problem Solving. *The thirty-second SIGCSE technical symposium on Computer Science Education (SIGCSE '01)*, pp. 21-25 .

Ginat, D., 2003. The greedy trap and learning from mistakes. *The 34th SIGCSE technical symposium on Computer science education (SIGCSE '03)*, pp. 11-15 .

Ginat, D., 2007. Hasty Design, Futile Patching and the Elaboration of Rigor. *The 12th annual SIGCSE conference on Innovation and technology in computer science education (ITiCSE '07)*, pp. 161 - 165 .

Ginat, D., 2008. Learning from Wrong and Creative Algorithm Design. *The 39th SIGCSE technical symposium on Computer science education(SIGCSE '08)*, pp. 26-30.

Gokhale, A. A., 1995. Collaborative Learning Enhances Critical Thinking. *Journal of Technology Education (JTE)*, 7(1), pp. 22-30.

González, M. P., Pascual, A. & Lorés, J., 2006. Evaluación heurística. *La Interacción Persona Ordenador*, pp. 1-40.

Goodrich, M. T. & Tamassia, R., 2001. *Data Structures and Algorithms in Java*. s.l.:John Wiley & Sons.

Guay, F., Vallerand, R. J. & Blanchard, C., 2000. On the assessment of situational intrinsic and extrinsic motivation: The Situational Motivation Scale (SIMS). *Motivation and Emotion*, Volumen 24, pp. 175-213.

Haberman, B., Averbuch, H. & Ginat, D., 2005. Is it really an algorithm- The need for explicit discourse. *The 10th annual SIGCSE conference on Innovation and technology in computer science education (ITiCSE '05)*, pp. 74 - 78 .

Hansen, S., Narayanan, N. H. & Hegarty, M., 2002. Designing Educationally Ejective Algorithm Visualizations. *Journal of Visual Languages and Computing* 13, 13(1), pp. 291-317.

Harter, S., 1981. new self-report scale of intrinsic versus extrinsic orientation in the classroom: motivational and informational components. *Developmental Psychology*, 17(3), pp. 300-312.

Herrmann, N., 1988. *The creative brain*. s.l.:Brain Books.

Herron, M., 1971. The nature of scientific inquiry. *School review*, pp. 171-212.

Higgins, C., Hegazy, T., Symeonidis, P. & Tsintsifas, A., 2003. The CourseMarker CBA System: Improvements over Ceilidh. *Education and Information Technologies*, 8(3), pp. 287 - 304.

Higgins, C., Symeonidis, P. & Tsintsifas, A., 2002. The Marking System for CourseMaster. *Seventh Annual Conference on Integrating Technology into Computer Science*, pp. 46-50.

Honey, P. & Mumford, A., 1986. *Using our Learning Styles*. s.l.:UK: Berkshire.

Horowitz, E. & Sahni, S., 1975. *Fundamentals of Computer Algorithms*. s.l.:Pitman.

Howley, I., Chaudhuri, S., Kumar, R. & Rosé, C., 2009. Motivation and Collaborative Behavior: An Exploratory Analysis. *CSCL'09 Proceedings of the 9th international conference on Computer supported collaborative learning*, Volumen 2, pp. 59-61 .

Hübscher-Younger, T. & Narayanan, N. H., 2003. Constructive and collaborative learning of algorithms. *34th SIGCSE Technical Symp, Computer Science Education, SIGCSE 2003.*, pp. 6-10.

Inaba, A. & Mizoguchi, R., 2004. Learners' Roles and Predictable Educational Benefits in Collaborative Learning An Ontological Approach to Support Design and Analysis of CSCL. *Intelligent Tutoring Systems*, Volumen 3220, pp. 285-294.

John, B. E. & Kieras, D. E., 1994. *The GOMS family of analysis techniques: Tools for design and evaluation*, s.l.: Human-Computer Interaction Institute Technical Report No. CMU-HCII-94-106..

Jurado, F., Molina, A. I., Redondo, M. A. & Cantero, M. O., 2012. Cole-Programming: Incorporando Soporte al Aprendizaje Colaborativo en Eclipse. *Revista Iberoamericana de Tecnologías del Aprendizaje (IEEE-RITA 7)*, 7(3), pp. 121-130.

Kaila, E., RAJALA, T., LAAKSO, M.-J. & SALAKOSKI, T., 2009. Effects, Experiences and Feedback from Studies of a Program Visualization Tool. *Informatics in Education*, Volumen 8, p. 17–34.

Keefe, J. W., 1979. Learning Style: An Overview". In NASSP's Student Learning Styles: Diagnosing and Prescribing Programs. *Reston, VA: National Association of Secondary School Principals*, pp. 1-17.

Khuri, S. & Holzapfel, K., 2001. EVEGA: An educational visualization environment for graph algorithms. *The 6th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE'01)*, pp. 101-104.

Knight, C. C. & Sutton, R. E., 2004. Neo-Piagetian Theory and Research: enhancing pedagogical practice for educators of adults. *London Review of Education*, 2(1), pp. 47-60.

Kolb, D. A., 1981. Learning Styles and Disciplinary Differences. *The modern American college*, pp. 233-255.

Kolb, D. A., 1984. *Experience as the source of learning and development*. Englewood Cliffs, NJ: Prentice-Hall.

Kolikant, Y. B.-D., 2005. Students' alternative standards for correctness. *The first International Workshop on Computing education research (ICER '05)*, pp. 37-43.

Kolikant, Y. B.-D. & Pollack, S., 2010. Establishing Computer Science Professional Norms among High-School Students. *Computer Science Education*, 14(1), pp. 21-35.

Koschmann, T., 1996. Paradigms shift and instructional technology. En: *Theory and practice of an emerging paradigm*. Mahwah NJ:: Lawrence Erlbaum Associates., pp. 1-23.

Kumar, A. J., Singhal, M. & Sheel, M. G., 2010. Educational Tool for Understanding Algorithm Building and Learning Programming Languages. *Proceedings of the World Congress of Engineering and Computer Science (WCES 2010)*, Volumen I, pp. 292-295.

Laxer, C., 2001. Treating computer science as science- An experiment with sorting. *The 6th annual conference on Innovation and technology in computer science education (ITiCSE '01)*, p. 189.

Leal, J. P. & Silva, F., 2003. Mooshak: a Web-based multi-site programming contest system. *Journal Software Practice & Experience*, 33(6), p. 567-581.

Lee, R. C. T., Tseng, S. S., Chang, R. C. & Tsai, Y. T., 2005. *Introducción al diseño y análisis de algoritmos*. s.l.:McGraw-Hill .

Lee, S., 1999. Usability Testing for Developing Effective Interactive Multimedia Software: Concepts, Dimensions, and Procedures. *Educational Technology & Society*, 2(2), pp. 1-13.

Levitin, A., 2000. Design and analysis of algorithms reconsidered. *The thirty-first SIGCSE technical symposium on Computer science education (SIGCSE '00)*, pp. 16-20.

Levitin, A., 2002. Using Puzzles in Teaching Algorithms. *The 33rd SIGCSE technical symposium on Computer science education*, pp. 292-296 .

Levitin, A., 2012. Digging for algorithmic Nuggets in the Land of Polyominoes. *The 17th ACM annual conference on Innovation and technology in computer science education (ITiCSE '12)*, pp. 52-56 .

Liu, D., Huang, S. & Brown, T., 2007. Supporting teaching and learning of optimisation algorithms with visualisation techniques. *In Proceedings of the 18th Conference of the Australasian Association for Engineering Education*, pp. 1-6.

Lucas, J., Naps, T. & Roessling, G., 2003. VisualGraph - A graph class designed for both undergraduate students and educators. *The 34th SIGCSE Technical Symposium on Computer Science Education (SIGCSE'03)*, pp. 167-171.

Malmi, L. y otros, 2004. Visual Algorithm Simulation Exercise System with Automatic Assessment: TRAKLA2. *Informatics in Education*, Volumen 33, pp. 267-288.

Marcelino, M., Mihaylov, T. & Mendes, A., 2008. H-SICAS, a Handheld Algorithm Animation And Simulation Tool To Support Initial Programming. *38th ASEE/IEEE Frontiers in Education Conference*, Volumen T4A, pp. 7-12.

Marcos, J. A. G. y otros, 2009. Detecting and Solving Negative Situations in Real CSCL Experiences with a Role-Based Interaction Analysis Approach. En: *Intelligent Collaborative e-learning Systems and Applications. Serie Studies in Computational Intelligence*. s.l.:Springer-Berlag, pp. 129-146.

Martín-Albo, J., Núñez, J. L. & J.G.Navarro, 2009. Validation of the Spanish Version of the Situational Motivation Scale (EMSI) in the Educational Context. *The Spanish Journal of Psychology*, 12(2), pp. 799-807.

Martínez, F. J. A., Urquiza-Fuentes, J. & Velázquez- Iturbide, J. Á., 2009. Visualization of Syntax Trees for Language Processing Courses. *Advanced Learning Technologies, 2009. ICALT 2009. Ninth IEEE International Conference on*, pp. 597-601.

Martínez, F. J. A., Urquiza-Fuentes, J. & Velázquez- Iturbide, J. Á., 2009. Visualization of Syntax Trees for Language Processing Courses. *Ninth IEEE International Conference on Advanced Learning Technologies (ICALT 2009)*, pp. 597 - 601.

- Matocha, J., 2002. Laboratory experiments in an algorithms course: technical writing and the scientific method. *Frontiers in Education*, Volumen 1, pp. 9-13.
- McCracken, D., 1989. Three “lab assignments” for an algorithms course. *ACM SIGCSE Bulletin*, 21(2), pp. 61-64.
- Michael, J., 2006. Where’s the evidence that active learning works?. *Adv Physiol Educ* 30, 30(4), p. 159–167.
- Moore, T. K., Rich, A. G. & Wick, M. R., 1993. Scientific investigation in a breadth-first approach to introductory computer science. *The twenty-fourth SIGCSE technical symposium on Computer science education (SIGCSE '93)*, 25(1), pp. 63-67.
- Moreno, A., Myller, N., Sutinen, E. & Ben-Ari, M., 2004. Visualizing programs with Jeliot 3. *In Proceedings of the International Working Conference on Advanced Visual Interfaces AVI 2004*, p. 373–376.
- Mugny, G. & Doise, W., 1978. Socio-cognitive conflict and structure of individual and collective performances. *European Journal of Social Psychology*, 8(2), p. 181–192.
- Naps, T., 2005. JHAVÉ: Supporting Algorithm Visualization. *IEEE Computer Graphics and Applications*, Volumen 25, pp. 49-55.
- Neapolitan, R. & Naimipour, K., 1999. *Foundations of Algorithms*. s.l.:Jones and Bartlett .
- Nielsen, J., 1992. Finding usability problems through heuristic evaluation. *SIGCHI Conference on Human Factors in Computing Systems*, pp. 373-380 .
- Nielsen, J., 2012. *Usability 101: Introduction to usability*. [En línea] Available at: <http://www.useit.com/alertbox/20030825.html> [Último acceso: 22 10 2013].
- Nuutinen, N. M. J., 2006. JeCo: Combining Program Visualization and Story Weaving. *Informatics in Education*, Volumen 5, p. 255–264.
- Oliet, N. M., Ortega, Y. & Verdejo, J. A., 1999. *Estructuras de datos y métodos algorítmicos: ejercicios resueltos*. s.l.:Pearson.
- Palincsar, A. S., 1998. Social constructivist perspectives on teaching and learning. *Annual Review of Psychology*, Volumen 49, pp. 345-375.

Papalaskari, M.-A., 2003. Peer evaluation in an algorithms course. *Proceedings of the 10th annual SIGCSE conference on Innovation and Technology in Computer Science Education (ITiCSE 2003)*, pp. 69-73.

Parberry, I., 1995. *Problems on Algorithms*. s.l.:Prentice Hall .

Pareja-Flores, C., Urquiza-Fuentes, J. & Velázquez-Iturbide, J. Á., 2007. WinHIPE: An IDE for functional programming based on rewriting and visualization. *ACM SIGPLAN Notices*, 42(3), pp. 14-23.

Pérez-Carrasco, A., Velázquez-Iturbide, J. & Urquiza Fuentes, J., 2010. Multiple usability evaluations of a program animation tool. *The 10th IEEE International Conference on Advanced Learning Technologies*, pp. 452-454.

Pérez, J. R. P., Puerto, M. R. P. & Cueva, J. M. L., 2006. SICODE: A collaborative tool for learning of software development. *IV International Conference on Multimedia and Information & Communication Technologies in Education*, pp. m-ICTE2006.

Perrenet, J., Groote, J. F. & Kaasenbrood, E., 2005 . Exploring Students' Understanding of the Concept of Algorithm: Levels of Abstraction. *The 10th annual SIGCSE conference on Innovation and technology in computer science education (ITiCSE '05)*, pp. 64 - 68 .

Prince, M., 2004. Does Active Learning Work? A Review of the Research. *Journal of Engineering Education*, Volumen 93, pp. 223-231.

Radosevic, D., Orehovacki, T. & Lovrencic, A., 2009. Verificator: Educational Tool for Learning Programming. *Informatics in Education*, Volumen 8, p. 261–280.

Ramani, K. & Rama Rao, T., 1994. A graphics based computer-aided learning package for integer programming: The branch and bound algorithm. *Computers & Education*, 23(4), pp. 261-268.

Revilla, M. A., Manzoor, S. & LIU, R., 2008. Competitive Learning in Informatics: The UVa Online Judge Experience. *Olympiads in Informatics*, Volumen 2, p. 131–148.

Rienties, B. y otros, 2012. The role of scaffolding and motivation in CSCL. *Computers & Education*, Volumen 59, pp. 893-906.

- Rienties, B. y otros, 2008. Students' motivations and their contributions to virtual learning. *ICLS'08 Proceedings of the 8th international conference on International conference for the learning sciences*, Volumen 2, pp. 254-261.
- Rienties, B. y otros, 2009. The role of academic motivation in Computer-Supported Collaborative Learning. *Computers in Human Behaviour*, 25(6), pp. 1195 - 1206.
- Riihiaho, S., 2009. User testing when test tasks are not appropriate. *ECCE '09 European Conference on Cognitive Ergonomics: Designing beyond the Product --- Understanding Activity and User Experience in Ubiquitous Environments*, p. Article No. 21 .
- Rittner, L. y otros, 2011. Adessowiki: Collaborative Scientific Programming Environment. *24th SIBGRAPI Conference on Graphics, Patterns, and Images Tutorials*, pp. 56 - 62.
- Röbling, G. & Freisleben, B., 2002. A System for Supporting Multiple Roles in Algorithm Animation. *Journal of Visual Languages and Computing*, Volumen 13, pp. 341-542.
- Röbling, G., Schüler, M. & Freisleben, B., 2000. The ANIMAL Algorithm Animation Tool. *The 5th annual SIGCSE/SIGCUE ITiCSE conference on Innovation and Technology in Computer Science Education (ITiCSE '00)*, pp. 37-40.
- Ryan, R. M. & Connell, J. P., 1989. Perceived locus of causality and internalization: examining reasons for acting in two domains. *Journal of Personality and Social Psychology*, 57(5), pp. 749-761.
- Ryan, R. M. & Deci, E. L., 2000. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology*, pp. 54-67.
- Sahni, S., 1995. *Data Structures, Algorithms, and Applications in Java*. s.l.:Silicon Press .
- Sánchez-Torrubia, M. G., Torres-Blanc, C. & Escribano-Blanco, S., 2010. An eLearning Environment Designed for Integrating eMathTeacher–Compliant Graph Algorithm Simulators. *Koli Calling '10 Proceedings of the 10th Koli Calling International Conference on Computing Education Research*, pp. 70-71.
- Sánchez-Torrubia, M., Torres-Blanc, C. & López-Martínez, M., 2009. PathFinder: A Visualization eMathTeacher for Actively Learning Dijkstra's Algorithm. *Electronic Notes in Theoretical Computer Science*, 224(1), pp. 151-158.

- Sánchez-Torrubia, M., Torres-Blanc, C. & López-Martínez, M., 2009. PathFinder: A Visualization eMathTeacher for Actively Learning Dijkstra's Algorithm. *Electronic Notes in Theoretical Computer Science*, pp. 151-158.
- Santos, Á., Gomes, A. & Mendes, A., 2010. Integrating New Technologies and Existing Tools to Promote Programming Learning. *Algorithms*. 3, pp. 183-196.
- Sayago, S., Navarrete, T. & Blat, J., 2003. Técnicas de Ingeniería de Usabilidad y metodología de diseño conceptual en algunas aplicaciones informáticas. *IV Congreso Internacional de Interacción-Persona Ordenador*.
- Sedgewick, R., 2002. *Algorithms in Java*. s.l.:Addison-Wesley.
- Serrano-Cámara, L., Paredes-Velasco, M., Alcover, C. & Velázquez-Iturbide, J., 2013. An evaluation of students' motivation in computer-supported collaborative learning of programming concepts. *Computers in Human Behavior*, Volumen 31, pp. 499-508.
- Shaffer, C. A. y otros, 2010. Algorithm Visualization: The State of the Field. *ACM Transactions on Computing Education (TOCE)*, 10(3), p. No. 9 .
- Sherstov, A., 2003. Distributed visualization of graph algorithms. *The 34th SIGCSE Technical Symposium on Computer Science Education (SIGCSE'03)*, pp. 376-380.
- Skiena, S., 2002. *The Algorithm Design Manual*. s.l.:Springer-Verlag .
- Sperry, R. W., 1961. Cerebral Organization and Behavior. *The American Association for the Advancement of Science*, Volumen Vol. 133, No. 3466, p. 1749 – 1757.
- Squire, D. & Preece, J., 1996. Usability and learning: Evaluating the potential of educational software. *Computers & Education*, 27(1), pp. 15-22.
- Sulaiman, J., Zulkifli, T., Ibrahim, K. & Noor, N. K. M., 2009. Implementing Usability In E-Learning System Using Hybrid Heuristics. *International conference on information and multimedia technology*, pp. 189-193.
- Tapola, A. y otros, 2001. Motivation and Participation in Inquiry Learning within a Networked Learning Environment. *European Perspectives on Computer-Supported Collaborative Learning. Proceedings of the First European Conference on CSCL*, pp. 585-592.

Urquiza-Fuentes, J. & Velázquez-Iturbide, J. Á., 2009. A Survey of Successful Evaluations of Program Visualization and Algorithm Animation Systems. *ACM Transactions on Computing Education (TOCE) - Special Issue on the 5th Program Visualization Workshop (PVW'08)*, 9(2), p. No. 9 .

Vallerand, R. J., 1997. Toward a hierarchical model of intrinsic and extrinsic motivation. *Advances in Experimental Social Psychology*, Volumen 29, p. 271–360.

Velázquez-Iturbide, J., 2011. The design and coding of greedy algorithms revisited. *16th Annual Conference on Innovation and Technology in Computer Science Education*, pp. 8-12.

Velázquez-Iturbide, J. Á., 2012. Refinement of an experimental approach to computer-based, active learning of greedy algorithms. *Innovation and Technology in Computer Science Education (ITiCSE)*, pp. 46-51.

Velázquez-Iturbide, J. Á., 2013. An Experimental Method for the Active Learning of Greedy Algorithms. *ACM Transactions on Computing Education*, 13(4), p. Article No. 18 .

Velázquez-Iturbide, J. Á., Aceptado. Identification and Removal of Misconceptions on Optimization Concepts Underlying Greedy Algorithms. *Journal of Research and Practice in Information Technology*.

Velázquez-Iturbide, J. Á. & Debdi, O., 2011. Experimentation with Optimization Problems in Algorithm Courses. *International Conference on Computer as a Tool (EUROCON)*, Volumen 2, pp. 1-4.

Velázquez-Iturbide, J. Á., Debdi, O., Esteban-Sánchez, N. & Pizarro, C., 2013a. GreedEx: A Visualization Tool for Experimentation and Discovery Learning of Greedy Algorithms. *IEEE Transactions on Learning Technologies*, Volumen 6, pp. 130-143.

Velázquez-Iturbide, J. Á. y otros, 2010. Un asistente extensible para la experimentación interactiva con problemas combinatorios. *XI Simposio Nacional de Tecnologías de la Información y las Comunicaciones en la Educación (SINTICE 2010)*, pp. 63-70.

Velázquez-Iturbide, J. Á., Martín-Torres, R. & González-Rabanal, N., 2013. OptimEx: un sistema para la experimentación con algoritmos de optimización. *XV International Symposium on Computers in Education (SIIE13)*, pp. 30-35.

Velázquez-Iturbide, J. Á., Paredes-Velasco, M. & Debdi, O., 2013c. GreedExCol: Una Herramienta Educativa Basada en CSCL para el Aprendizaje de Algoritmos Voraces. *XIV Simposio Nacional de Tecnologías de la Información y las Comunicaciones en la Educación (SINTICE 2013)*, pp. 96-103.

Velázquez-Iturbide, J. Á., Pareja-Flores, C., Debdi, O. & Paredes-Velasco, M., 2012. Interactive Experimentation with Algorithms. En: S. Abramovich, ed. *Computers in Education – Volumen 2*. s.l.:Nova, pp. 47-70.

Velázquez-Iturbide, J. Á. & Pérez-Carrasco, A., 2009a. Active learning of greedy algorithms by means of interactive experimentation. *14th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, pp. 119-123.

Velázquez-Iturbide, J. Á., Pérez-Carrasco, A. & Debdi, O., 2013b. Experiences in Usability Evaluation of Educational Programming Tools. En: C. González, ed. *Student Usability in Educational Software and Games: Improving Experiences*. s.l.:Carina González (ed.), IGI Global, pp. 241-260.

Velázquez-Iturbide, J. Á., Pérez-Carrasco, A. & Urquiza-Fuentes, J., 2009b. Interactive visualization of recursion with SRec. *The 14th annual ACM SIGCSE conference on Innovation and technology in computer science education (ITiCSE '09)*, 41(3), pp. 339-339.

Velázquez-Iturbide, J., Lázaro-Carrascosa, C. & Hernán Losada, I., 2009c. Asistentes interactivos para el aprendizaje de algoritmos voraces. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje (IEEE-RITA)*, 4(3), pp. 213-220.

Vijay, J. S. D. y otros, 2008. Jazz Sangam: A Real-time Tool for Distributed Pair Programming of a Team Development Platform. *Infrastructure for Research on Collaborative Software Engineering (IRCSE)*.

Virtanen, A., Lahtinen, E. & Jarvinen, H. M. V., 2005. A visual interpreter for learning introductory programming with C++. *In Proceedings of the 5th Koli Calling Conference on Computer Science Education (KOLI'05)*, p. 125–130.

Vizcano, A., Contreras, J., Favela, J. & Prieto, M., 2000. An Adaptive, Collaborative Environment to Develop Good Habits in Programming. *Intelligent Tutoring Systems, Lecture Notes in Computer Science*, Volumen 1839, pp. 262-271.

Walker, H. M., 1997 . A racquetball or volleyball simulation. *ACM SIGCSE Bulletin*, 29(4), pp. 22-23 .

Wharton, C., Rieman, J., Lewis, C. & Polson, P., 1994. The cognitive walkthrough method: A practitioner's guide. En: J. Nielsen & R. Mack, edits. *Usability inspection methods*. s.l.:John Wiley & Sons, pp. 105-140.

WikiBooks, 2014. <https://en.wikibooks.org/>. [En línea] Available at: https://en.wikibooks.org/wiki/Computer_Science_Design_Patterns/Model%20%80%93view%20%80%93controller [Último acceso: 19 05 2014].

Winslow, L. E., 1996 . Programming Pedagogy -- A Psychological Overview. *ACM SIGCSE Bulletin*, 28(3), pp. 17-22 .

Wu, M., 2005. Teaching graph algorithms using online Java package IAPPGA. *ACM SIGCSE Bulletin*, 37(4), pp. 64-68.

Zachary, J. L., 1997. The Gestalt of Scientific Programming: Problem, Model, Method, Implementation, Assessment. *The twenty-eighth SIGCSE technical symposium on Computer science education (SIGCSE '97)*, pp. 238-242.

Zywno, S., 2003. *A Contribution to Validation of Score Meaning for Felder-Soloman's Index of Learning Styles*. s.l., Ryerson University.

