



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERÍA INFORMÁTICA

Curso Académico 2010/2011

Proyecto de Fin de Carrera

**EDICIÓN Y PUBLICACIÓN
DE
PRESENTACIONES MULTIMEDIA
ADAPTATIVAS
CON FINES EDUCATIVOS**

Autores: María Asunción Pastor Moya

Luis Miguel García Olmos

Tutor: Jaime Urquiza Fuentes

*A todos aquellos
que siempre han estado ahí.*

AGRADECIMIENTOS

Queremos expresar nuestro agradecimiento más sincero a todos aquellos que han hecho posible este proyecto.

En primer lugar, queremos agradecer muy especialmente a Jaime Urquiza Fuentes, nuestro tutor, su apoyo y paciencia todo este tiempo.

Yo, María Asunción, a mis padres y a mi hermana, en general a toda mi familia, por su apoyo incondicional. A mi compañero Luis Miguel porque sin él este proyecto no estaría completo.

Yo, Luis Miguel, a toda mi familia y en particular a mi hermana Irene. A mi compañera por encauzar el proyecto hacia la buena dirección.

A todo el software libre en el que se basa este proyecto.

Por último, a nuestros amigos, por darnos ánimos en tiempos difíciles.

GRACIAS,

RESUMEN

Dentro del ámbito de la enseñanza existen muchos métodos y técnicas que constituyen los recursos necesarios, y permiten adquirir las habilidades y aptitudes necesarias con el menor esfuerzo.

Hoy día, gracias a la presencia de los recursos tecnológicos en nuestra vida diaria, estos métodos y técnicas han evolucionado dando lugar a la enseñanza online/virtual. Con ella ha aparecido el término de e-Learning, que pretende aprovechar las nuevas tecnologías para ayudar a los alumnos a estudiar.

Por tanto, e-Learning sería la utilización de todo tipo de herramientas informáticas por parte del docente, siendo una de las más conocidas Internet [30]. Una de las ventajas es que facilita al alumnado el acceso a contenidos y materiales formativos en cualquier momento, e independientemente del lugar en que se encuentre, eliminando las dificultades de la docencia presencial.

Así, este proyecto se ha basado en e-Learning, ya que ha pretendiendo crear documentos interactivos con contenido formativo. Convirtiéndose en una herramienta con la que el alumno pueda aprender de una forma más estimulante e incentivadora, y consiga sus objetivos.

Para conseguirlo, este proyecto se ha dividido en dos partes:

- Por un lado, la creación de una herramienta que permita al docente diseñar documentos con contenido formativo, de una manera amena y sencilla.
- Y, por otro lado, la creación de una aplicación web, donde el docente pueda gestionar tanto estos documentos interactivos como sus asignaturas y alumnos. Donde los alumnos podrán descargarse y visualizar las presentaciones secuenciales o adaptativas, según la preferencia del docente.

La herramienta de edición se compondrá de dos vistas:

- Diseño de un diagrama de nodos (grafo), donde el docente decidirá la navegación entre todos los nodos del documento. De tal forma que, el alumno adquiera nuevos conocimientos o consolide los ya aprendidos, ayudando así a su aprendizaje.
- Diseño del contenido de cada nodo (paso), donde el profesor añadirá información, gracias a la inclusión de textos (componente narrativo), medios (componente multimedia) y preguntas (componente evaluador).

Además, este editor permitirá la modificación de documentos creados anteriormente. Teniendo la posibilidad de o bien exportar el documento creado, para su importación en una aplicación web, o bien la generación de la presentación final, que podrá ser vista mediante un navegador web.

Por su parte, la aplicación web, además de lo dicho anteriormente, permitirá al docente la generación offline y online de la presentación final. Su descarga y visualización por el alumno, desde cualquier sitio y a cualquier hora.

La generación de esta presentación interactiva se basa en la especificación de los distintos pasos que contendrá dicha presentación. Cada paso tendrá definidos una serie de componentes: narrativo, multimedia y evaluador, con los que el alumno irá adquiriendo conocimientos de una manera más atractiva. Gracias a una serie de reglas definidas por el docente irá adquiriendo nuevas nociones o asentará las ya aprendidas.

Por último, destacaría la importancia de la posibilidad que tiene dicha presentación para adaptarse a los conocimientos del estudiante. De tal manera, que si el docente elige una generación adaptativa, el documento tendrá que mostrar al alumno unos pasos u otros, dependiendo del nivel que tenga el alumno, en ese momento. El alumno, sólo podrá llegar al final, si ha adquirido los conocimientos suficientes.

Índice

1. INTRODUCCIÓN	1
1.1. OBJETIVOS	1
2. ESTADO DE LA CUESTIÓN	3
2.1. e-LEARNING	3
2.2. Editores	3
2.3. Repositorios	4
3. DESCRIPCIÓN INFORMÁTICA	7
3.1. REQUISITOS	7
3.1.1. Descripción General	7
3.1.2. Requisitos funcionales	9
3.1.3. Requisitos no funcionales	12
3.1.4. Diagrama de Casos de uso	13
3.2. ANÁLISIS	17
3.2.1. Diagramas de colaboración de los Casos de uso	17
3.3. DISEÑO	51
3.3.1. Diagrama de clases: Modelo de Datos del documento	51
3.3.2. Diagrama de clases: Modelo de Datos de las estructuras HTML	60
3.3.3. Diagrama de clases: Editor	62
3.3.4. Diagrama de clases: Repositorio y Librería generador	69
3.3.5. Diseño de la Base de Datos	79
3.3.5.1. Diagramas Entidad-Relación	79
3.3.5.2. Esquemas Entidad-Relación	84
3.4. IMPLEMENTACIÓN	87
3.4.1. Diferencias: requisitos y resultado	89
3.4.1.1. Requisitos iniciales del sistema	89

3.4.1.2. Resultado final	91
3.4.2. Plataforma	92
3.4.2.1. Editor	92
3.4.2.2. Repositorio	93
3.4.2.3. Generación y visualización de documentos	93
3.5. PRUEBAS	101
3.5.1. Ejemplo de funcionamiento	105
3.5.1.1. Editor	105
3.5.1.2. Repositorio	115
3.5.1.3. Visualización de documentos	122
4. CONCLUSIONES	127
4.1. Evaluación	127
4.2. Trabajos futuros	129
5. CONTENIDO DEL CD-ROM	131
A. MANUAL DE USUARIO: Editor	I
B. MANUAL DE USUARIO: Repositorio	XIII
C. LENGUAJES DE MARCADO	XIX
Glosario	XXIII
Bibliografía	XXIX

Índice de figuras

3.1. Caso de uso: crear nuevo documento y diagrama	13
3.2. Caso de uso: añadir pasos, reglas y validar la navegación	13
3.3. Caso de uso: añadir componentes	14
3.4. Caso de uso: gestión de documentos	14
3.5. Caso de uso: gestión de grupos	14
3.6. Caso de uso: asociación de grupos y documentos	15
3.7. Caso de uso: asociación de grupos y alumnos	15
3.8. Caso de uso: descarga de documentos offline	15
3.9. Caso de uso: gestión de usuarios	15
3.10. Caso de uso: generación de documentos offline	16
3.11. Caso de uso: generación de documentos online	16
3.12. Caso de uso: visualización de documentos offline	16
3.13. Caso de uso: visualización de documentos online	16
3.14. Diagrama de colaboración de <i>Crear documento y diagrama de navegación</i>	17
3.15. Diagrama de colaboración de <i>Añadir paso</i>	18
3.16. Diagrama de colaboración de <i>Editar paso</i>	19
3.17. Diagrama de colaboración de <i>Editar componentes del paso</i>	20
3.18. Diagrama de colaboración de <i>Añadir regla de navegación</i>	21
3.19. Diagrama de colaboración de <i>Editar regla</i>	22
3.20. Diagrama de colaboración de <i>Editar estado de conocimiento de la regla</i>	23
3.21. Diagrama de colaboración de <i>Validar navegación</i>	24
3.22. Diagrama de colaboración de <i>Añadir componente narrativo</i>	25
3.23. Diagrama de colaboración de <i>Editar componente narrativo</i>	26
3.24. Diagrama de colaboración de <i>Borrar componente narrativo</i>	27
3.25. Diagrama de colaboración de <i>Añadir componente multimedia</i>	28
3.26. Diagrama de colaboración de <i>Editar componente multimedia</i>	29
3.27. Diagrama de colaboración de <i>Borrar componente multimedia</i>	30

3.28. Diagrama de colaboración de <i>Añadir componente evaluador</i>	31
3.29. Diagrama de colaboración de <i>Editar componente evaluador</i>	32
3.30. Diagrama de colaboración de <i>Editar estado de conocimiento del componente evaluador</i>	34
3.31. Diagrama de colaboración de <i>Borrar componente evaluador</i>	35
3.32. Diagrama de colaboración de <i>Incluir archivo</i>	36
3.33. Diagrama de colaboración de <i>Eliminar archivo</i>	37
3.34. Diagrama de colaboración de <i>Crear Grupo</i>	38
3.35. Diagrama de colaboración de <i>Eliminar Grupo</i>	39
3.36. Diagrama de colaboración de <i>Editar Grupo</i>	40
3.37. Diagrama de colaboración de <i>Asociar grupos y documentos</i>	41
3.38. Diagrama de colaboración de <i>Asociar grupos y alumnos</i>	42
3.39. Diagrama de colaboración de <i>Descargar documento offline</i>	43
3.40. Diagrama de colaboración de <i>Crear usuarios</i>	44
3.41. Diagrama de colaboración de <i>Crear usuarios</i>	45
3.42. Diagrama de colaboración de <i>Modificar usuarios</i>	46
3.43. Diagrama de colaboración de <i>Generar Offline secuencial o adaptativa</i> .	47
3.44. Diagrama de colaboración de <i>Generar Online secuencial o adaptativa</i> .	48
3.45. Diagrama de colaboración de <i>Visualizar documento online</i>	49
3.46. Diagrama de colaboración de <i>Visualizar documento offline</i>	50
3.47. Diagrama de clases de <i>Documento</i>	51
3.48. Diagrama de clases de <i>Paso</i>	52
3.49. Diagrama de clases de <i>Regla</i>	52
3.50. Diagrama de clases de <i>Componentes: narrativo, multimedia y evaluador</i>	53
3.51. Diagrama de clases de <i>Parametro</i>	54
3.52. Diagrama de clases de <i>Enunciado</i>	54
3.53. Diagrama de clases de <i>Parrafo</i>	54
3.54. Diagrama de clases de <i>TextoCss</i>	55
3.55. Diagrama de clases de <i>Pista</i>	55
3.56. Diagrama de clases de <i>Preguntas: test y valor</i>	56
3.57. Diagrama de clases de <i>Respuestas: test y valor</i>	57
3.58. Diagrama de clases de <i>Feedback</i>	57
3.59. Diagrama de clases de <i>Conocimiento</i>	58
3.60. Diagrama de clases de <i>EstadoDeConocimiento</i>	58
3.61. Diagrama de clases de <i>ConocimientoException</i>	58
3.62. Diagrama de clases de <i>ErrorNavegacion</i>	59

3.63. Diagrama de clases de <i>CadenaErroresNavegacion</i>	59
3.64. Diagrama de clases de <i>DocumentRoot</i>	60
3.65. Diagrama de clases de <i>Estructuras</i> y <i>Estructura</i>	61
3.66. Diagrama de clases de <i>ParamType</i>	61
3.67. Diagrama de clases de <i>BaseEdicion</i> , <i>RespuestaEdicion</i> (test/valor) y <i>ParrafoEdicion</i>	62
3.68. Diagrama de clases de <i>CNarrativoDialog</i> , <i>CMultimediaDialog</i> y <i>CEvaluadorDialog</i>	63
3.69. Diagrama de clases de <i>ParametroEdicion</i> y <i>ValorParametroEditingSupport</i>	63
3.70. Diagrama de clases de <i>EditarTextoDialog</i> , <i>EditorTextoConCssDialog</i> y <i>EditorParrafoDialog</i>	64
3.71. Diagrama de clases de <i>TextoLabelProvider</i> y <i>CssLabelProvider</i>	64
3.72. Diagrama de clases de <i>EditorRespuestaTestDialog</i> y <i>EditorRespuestaValorDialog</i>	65
3.73. Diagrama de clases de <i>ConceptoEdicion</i>	65
3.74. Diagrama de clases de <i>ConceptosTreeContentProvider</i> y <i>ConceptoTreeLabelProvider</i>	66
3.75. Diagrama de clases de <i>ConceptoDragListener</i> y <i>ConceptoDropAdapter</i>	66
3.76. Diagrama de clases de <i>ConceptoValidator</i>	66
3.77. Diagrama de clases de <i>FeedBackEdicion</i> y <i>EditorFeedBackDialog</i>	67
3.78. Diagrama de clases de <i>ConocimientoEdicion</i> y <i>EditorConocimientoDialog</i>	67
3.79. Diagrama de clases de <i>EstadoDeConocimientoDialog</i>	68
3.80. Diagrama de clases de <i>Forms</i>	69
3.81. Diagrama de clases de <i>Actions</i>	70
3.82. Diagrama de clases de <i>POJOs</i>	71
3.83. Diagrama de clases de <i>DAOs</i>	71
3.84. Diagrama de clases de <i>AlmacenPropiedades</i>	72
3.85. Diagrama de clases de <i>Security</i>	72
3.86. Diagrama de clases de <i>Validation</i>	72
3.87. Diagrama de clases de <i>GeneradorDocumento</i> , <i>GeneradorPaso</i> y <i>GeneradorHTML</i>	73
3.88. Diagrama de clases de <i>PaginaWeb</i> y <i>PaginaWebOnline</i>	74
3.89. Diagrama de clases de <i>AlmacenPropiedades</i> y <i>AlmacenEstructuras</i>	75
3.90. Diagrama de clases de <i>Utils</i> , <i>ElementoHTML</i> y <i>ResourceDocumento</i>	75
3.91. Diagrama de clases de <i>FaltaPropiedadException</i> y <i>Messages</i>	76
3.92. Diagrama de clases de <i>IDocumentoDAO</i>	76

3.93. Diagrama de clases de <i>WrapperDoc</i> , <i>WrapperGrupo</i> y <i>WrapperDocAlumno</i>	77
3.94. Diagrama de clases de <i>DController</i>	77
3.95. Diagrama de clases de <i>Utils</i> y <i>GestionDocumentosException</i>	78
3.96. Diagrama de E/R del repositorio : usuario <i>Admin</i>	79
3.97. Diagrama de E/R del repositorio : usuario <i>Alumno</i>	80
3.98. Diagrama de E/R del repositorio : usuario <i>Profesor</i>	82
3.99. Diagrama de E/R de la <i>generación de documentos offline</i>	82
3.100 Diagrama de E/R de la <i>generación de documentos online</i>	83
3.101 Diagrama de E/R de la <i>visualización de documentos online</i>	83
3.102 Diagrama de E/R de documentos, versiondocumentos y grupos	84
3.103 Diagrama de E/R de grupos y alumnos	85
3.104 Diagrama de E/R de documentos offline y online secuenciales	86
3.105 Diagrama de E/R de documentos offline y online adaptativos	86
3.106 Primera pantalla de la aplicación.	106
3.107 Diálogo para crear un nuevo documento y diagrama de navegación.	106
3.108 Diálogo para editar documento de estructuras.	107
3.109 Diálogo para editar documento de estilo.	107
3.110 Diálogo para crear un nuevo documento y diagrama de navegación.	108
3.111 Diálogo para editar propiedades de los pasos y reglas de navegación.	108
3.112 Diálogo para validar navegación.	109
3.113 Diálogo para añadir componentes: narrativo, multimedia y evaluador.	109
3.114 Diálogo para editar componente narrativo.	110
3.115 Diálogo para editar componente multimedia.	111
3.116 Diálogo para editar componente evaluador.	112
3.117 Diálogo para editar el estado de conocimiento del componente evaluador.	113
3.118 Diálogo para exportar a la Web.	113
3.119 Diálogo para generar un documento offline.	114
3.120 Pantalla de acceso a la aplicación Web	115
3.121 Pantalla: Gestión Documentos	116
3.122 Selección del documento .zip	116
3.123 Subir documento al Servidor	116
3.124 Generar documento offline u online	117
3.125 Pantalla: Gestión Grupos	118
3.126 Crear grupo	118
3.127 Pantalla: Asoc. Grupos y Documentos	119
3.128 Asociar documento a grupo/s	119

3.129	Pantalla: Asoc. Grupos y Alumnos	120
3.130	Asociar documento a alumno/s	120
3.131	Pantalla: Ver documentos	121
3.132	Pantalla: Gestión Usuarios	121
3.133	Paso 1: Componente narrativo, multimedia (video YouTube) y evaluador (test)	122
3.134	Paso 2: Componente multimedia (imagen) y evaluador (valor único) . .	122
3.135	Paso 3: Componente multimedia (imagen) y evaluador (test)	123
3.136	Paso 4: Componentes multimedia (Flash e imagen) y evaluador (valor: rellenar huecos)	123
3.137	Paso 5: Componente narrativo, multimedia (imagen) y evaluador (valor: rellenar huecos)	124
3.138	Paso 6: Componente narrativo, multimedia (video .avi y sonido) y evaluador (test)	124
3.139	Paso 7: Componente multimedia (video .mov e imagen) y evaluador (valor: rellenar huecos)	125
3.140	Ejemplo: Resultado de Visualización offline/online	125
A.1.	Primera pantalla de la aplicación.	I
A.2.	Diálogo para crear un nuevo documento y diagrama de navegación. . .	II
A.3.	Diálogo para crear un nuevo documento y diagrama de navegación. . .	II
A.4.	Diálogo para editar propiedades de los pasos y reglas de navegación. . .	III
A.5.	Diálogo para editar el estado de conocimiento de la regla	IV
A.6.	Diálogo para añadir componentes: narrativo, multimedia y evaluador. .	IV
A.7.	Diálogo para editar las <i>propiedades</i> del componente evaluador.	V
A.8.	Diálogo para editar el estado de conocimiento del componente evaluador.	V
A.9.	Menú contextual: Modificar.	VI
A.10.	Resultado de la edición del componente evaluador.	VII
A.11.	Diálogo para editar las <i>propiedades</i> del componente multimedia.	VIII
A.12.	Menú contextual: Modificar.	VIII
A.13.	Resultado de la edición del componente multimedia.	IX
A.14.	Diálogo para editar las <i>propiedades</i> del componente narrativo.	IX
A.15.	Menú contextual: Modificar.	X
A.16.	Resultado de la edición del componente narrativo.	X
A.17.	Diálogo para exportar a la Web.	XI
A.18.	Diálogo para generar un documento offline.	XI
A.19.	Diálogo para editar documento de estructuras.	XII

A.20. Diálogo para editar documento de estilo	XII
B.1. Pantalla de acceso a la aplicación Web	XIII
B.2. Pantalla: Gestión Documentos	XIV
B.3. Eliminar documento	XIV
B.4. Generar documento offline u online	XV
B.5. Pantalla: Gestión Grupos	XVI
B.6. Pantalla: Asoc. Grupos y Documentos	XVI
B.7. Pantalla: Asoc. Grupos y Alumnos	XVII
B.8. Pantalla: Ver documentos	XVII
B.9. Pantalla: Gestión Usuarios	XVIII

INTRODUCCIÓN

1.1. OBJETIVOS

Como ya se ha puntualizado, el proyecto se encuadra dentro del campo de la educación virtual/online. En concreto, se pretende encuadrar en el ámbito de e-Learning, que gracias al avance de las nuevas tecnologías, poco a poco se ha ido introduciendo tanto en las universidades como en las empresas.

Desde la antigüedad, el hombre ha tenido la necesidad de enseñar o compartir sus conocimientos con la sociedad. Los sistemas de educación conocida más antiguos enseñaban religión y mantenían las tradiciones. Por ejemplo, en Egipto, en las escuelas se enseñaba además de religión, los principios de la escritura, ciencia, matemáticas y arquitectura. Poco a poco, la educación fue evolucionando hasta lo que hoy se conoce.

En la actualidad, gracias al avance de las nuevas tecnologías, al alcance de todos, aparte de la evolución de los métodos y técnicas para la formación presencial, se ha querido ir más allá. Apareciendo una nueva forma de enseñanza que quiere salvar las dificultades que conlleva la formación presencial. De tal manera que el alumno pueda interactuar con el profesor, participar de forma activa independientemente del lugar donde se encuentre, u obtener el material formativo en el momento.

Por tanto, el objetivo principal del uso de las nuevas tecnologías en la educación es el de prestar servicio a los alumnos. Mejorando el acceso al contenido y material formativo, y ayudando a la asimilación de nuevas habilidades o aptitudes ante la/s asignatura/s o conocimiento/s que se les planteen.

El objetivo principal de este proyecto es la edición y generación de documentos

formativos para e-Learning. Estos documentos formativos tienen el fin de conseguir que el alumno avance en la asimilación de conceptos dentro de cada asignatura, de manera más dinámica. El alumno obtiene mucha información en cada paso del documento, y solo si la ha entendido, obtendrá el nivel deseado. El docente, tiene la posibilidad de hacer que, esta información que recibe el estudiante, se adapte a lo que sabe (navegación adaptativa). Siendo algo novedoso en este tipo de herramientas de enseñanza online.

Este proyecto deberá, por una lado, conseguir la creación y edición de estos documentos:

1. Creando la navegación entre los diferentes pasos, de tal manera que el alumno vaya adquiriendo conocimiento.
2. Elaborando cada paso, añadiéndole funcionalidad: componentes narrativos, multimedia y, opcionalmente, un componente evaluador.

Y, por otro lado, generar las presentaciones interactivas finales, por medio de una serie de procesos, que el docente puede ejecutar a través del repositorio. Donde, también, el alumno podrá:

- o bien descargarlas en su equipo, como un archivo comprimido .zip (offline).
- o bien visualizarlas en el repositorio, actualizando sus conocimientos en una base de datos (online).

El resto de la memoria se estructura como sigue. En el capítulo 2 se realiza una introducción a e-Learning, además de hacer un repaso a los distintos editores y repositorios relacionados. En el capítulo 3 se realiza una descripción informática detallada del editor, repositorio y la generación y visualización de documentos formativos, incluyendo pruebas de funcionamiento sobre cada uno de ellos. Finalmente, en el capítulo 4 se exponen las conclusiones obtenidas del desarrollo del PFC, así como sus posibles trabajos futuros.

ESTADO DE LA CUESTIÓN

2.1. e-LEARNING

El aprendizaje electrónico, más conocido como e-Learning, es una modalidad de enseñanza diferente, que pretende cambiar la forma de aprender, a través del uso de herramientas que mejoran el desempeño de las personas [2].

Hoy día, un ejemplo claro de E-Learning es el “*Campus Virtual*” que tienen las universidades, por medio del cual todo alumno puede acceder al contenido formativo que desee, en cualquier momento y desde cualquier parte del Mundo, solo con conectarse a Internet. Con otras ventajas como: reducción de costes, no se requiere software específico, por parte del alumno, ni grandes conocimientos de Informática, no existen límites geográficos y el docente tiene flexibilidad horaria.

Además, de todo lo dicho anteriormente, e-Learning también es una herramienta de formación en los países menos desarrollados. Esto es, gracias a que permite su implantación en un plazo corto y con inversiones mucho menores a las que se necesitarían si la formación fuera presencial [7]. No solo esto, sino que también está siendo eficaz en sectores sociales como personas discapacitadas, desempleados, etc.

2.2. Editores

Existen muchas aplicaciones para e-Learning gratuitas como: HotPotatoes, JClic, Exelearning, CourseLab, Freemind o EdiLim [5], y otras de carácter comercial como: Sigmer o Articulate.

Este proyecto surge de la carencia de una herramienta, que proporcione al docente lo que realmente necesita, sin tener que pagar por ello. Las herramientas antes mencionadas, no cubren esta necesidad, están incompletas, o tienen funcionalidades que realmente no se utilizan. Así HotPotatoes y JClic están enfocados a producir ejercicios de distintos tipos, mientras que Exelearning va más hacia la generación de websites. Por otro lado, CourseLab y Edilim permiten crear material interactivo (cursos o actividades), mientras que Freemind permite crear mapas conceptuales, para estructurar proyectos o apuntes. Tanto HotPotatoes, JClic, CourseLab como Edilim nos permiten crear presentaciones interactivas, donde el alumno puede mejorar en sus conocimientos, pero ninguna de ellas se adapta al conocimiento del alumno.

El editor, que aquí se presenta, permite al profesor crear documentos portables, que se podrán descargar o estar disponibles desde cualquier repositorio Web. Además, permite la inclusión de cualquier tipo de pregunta, video (flash, .avi,...), imagen y sonido. Permitiendo la actualización de su contenido Web, pudiendo ser visualizable en el futuro, al ir evolucionando los navegadores Web. Teniendo una característica muy importante, se adapta al aprendizaje del alumno, permitiéndole adquirir conocimiento de una forma progresiva.

2.3. Repositorios

Existen muchos repositorios Web para e-Learning como: Moodle, Efront, Dokeos, Claroline o Ilias, todos ellos para aprendizaje virtual que permiten añadir ficheros, evaluar alumnos o la interacción entre alumnos y profesores [3].

El repositorio, que aquí se presenta, se diferencia de ellos que permite gestionar documentos, asignaturas y alumnos, y además generar presentaciones formativas con conocimientos asociados. Estas presentaciones tienen la posibilidad de adaptarse al conocimiento del alumno, a diferencia de otras herramientas. Una de ellas, líder mundial en el sector, es ToolBook, que permite generar también todo tipo de presentaciones interactivas con texto, sonido, video, Java Aplets o preguntas, pero no guarda el conocimiento de los alumnos y se adecúa a ellos. Esta es, sin duda, la característica más importante de la librería Java de generación, y de la base del proyecto.

Existen otras herramientas, de propósito general, como PowerPoint o similares, que permiten crear presentaciones interactivas, donde se pueden insertar ficheros

multimedia y preguntas, de una manera fácil y sencilla. Pero siguen teniendo la misma carencia explicada anteriormente, no se adaptan al aprendizaje del sujeto.

Además de lo comentado anteriormente, este conocimiento obtenido se guarda en base de datos. Pudiéndose ser utilizado, por el docente, para amoldarse a la velocidad de aprendizaje de los alumnos o tratar de mejorar sus métodos o técnicas de enseñanza. Además, en un futuro, podrá obtenerse de otros repositorios Web, y utilizarse también en el repositorio actual, consiguiendo avanzar en el conocimiento de sus alumnos, sin tener que empezar desde cero.

DESCRIPCIÓN INFORMÁTICA

En este capítulo se va a realizar una descripción general del editor, repositorio, y la generación y visualización de documentos formativos, enumerándose los requisitos funcionales y no funcionales. Además de mostrar los distintos casos de uso definidos.

3.1. REQUISITOS

En este apartado se realiza una descripción general del editor, el repositorio, y la generación y visualización de documentos formativos, enumerándose los requisitos funcionales y los no funcionales.

3.1.1. Descripción General

Editor

El editor debe ser capaz de crear documentos formativos desde cero, y también poder modificarlos. Para ello, por un lado, debe permitir la creación de las reglas de navegación y, por otro lado, la creación de los diferentes pasos con todo su contenido.

También tendrá que poder exportar un documento para su posterior importación en una aplicación Web. Además de la generación de un documento final, para su visualización en un navegador Web.

Por otro lado, este editor debe ser independiente del sistema operativo en el que se ejecute. Siempre teniendo en cuenta que tiene que dar soporte a la máquina virtual de Java.

Repositorio

A grandes rasgos, el repositorio debe ser capaz de:

- Para el docente: gestionar documentos, gestionar grupos, asociar grupos y documentos, y grupos a alumnos.
- Para el alumno: mostrar documentos (offline y online).
- Y para el administrador: gestionar usuarios.

No tendrá dependencia del sistema operativo en el que se ejecuta, salvo que debe también, el PC del alumno, dar soporte a la maquina virtual de Java.

Con respecto a la generación offline y online, el repositorio, en el apartado de *Gest. Documentos*, permitirá añadir los documentos. Estos, una vez cargados en el servidor, podrán ser utilizados para su generación offline y online.

Generación y visualización de documentos

La generación de documentos tiene que ser capaz, tanto a nivel offline como online, de crear todos los pasos que constituyen el documento final. Estos pasos llevarán consigo componentes: narrativo, multimedia y, de forma opcional, un evaluador, con un estado de conocimiento. Además tendrán asociadas unas reglas de navegación que, dependiendo del nivel de conocimiento, indicarán el paso siguiente a transitar.

La visualización de estos documentos debe ser capaz de mostrar cada paso, interactuar con el alumno, transitar al paso secuencial/adaptativo siguiente, dependiendo de como haya sido su generación, reproducir los archivos multimedia, y mostrar el resultado final del documento.

3.1.2. Requisitos funcionales

Los requisitos funcionales definen el comportamiento específico del software, sus funcionalidades, es decir, cómo los casos de uso serán llevados a la práctica.

Editor

El editor deberá permitir el siguiente funcionamiento.

RFE01 Será posible construir un documento desde cero, pudiendo ser realizada su modificación en cualquier momento.

RFE02 La herramienta permitirá la edición del documento de estructuras, para los componentes multimedia.

RFE03 La herramienta permitirá el cambio del estilo del documento.

RFE04 La herramienta poseerá dos vistas:

RFE04.01 Una vista para la edición de los pasos, su composición.

RFE04.02 Otra vista para organizar la navegación entre los distintos pasos.

RFE05 La modificación podrá hacerse de ambas vistas.

RFE06 El documento se compondrá de pasos.

RFE07 Cada paso podrá tener: uno o más componentes multimedia y narrativos, pero sólo un/ningún evaluador.

RFE07.01 El componente multimedia podrá contener una imagen, sonido o vídeo.

RFE07.02 El componente narrativo sólo podrá contener párrafos con texto.

RFE07.03 El componente evaluador podrá contener preguntas tipo test, un valor o rellenar huecos.

RFE07.04 El componente evaluador mostrará un *feedback* si la respuesta no ha sido correcta, con información referente a la pregunta, pero no dirá cual era la respuesta correcta.

RFE08 El orden de reproducción (o navegación) podrá ser secuencial o adaptativo:

RFE08.01 El orden secuencial es el definido en la secuencia de pasos del documento, sin importar la respuesta dada.

RFE08.02 El orden adaptativo es el que depende de las reglas de navegación, dependiendo del nivel obtenido del alumno nos devolverá el paso con nivel más próximo (igual o menor).

RFE09 En cualquier momento, se podrá validar la navegación, y con ella todo el documento, que devolverá un listado de errores encontrados.

RFE10 La herramienta permitirá la exportación del documento comprimido, con todos los archivos necesarios para su importación en una aplicación Web.

RFE11 La herramienta también permitirá la generación del documento formativo final, que se podrá ver mediante un navegador Web.

Repositorio

En el repositorio se enumeran los siguientes requisitos funcionales, en función del perfil del usuario:

■ Docente

RFW01 Será posible gestionar los documentos:

RFW01.01 Se permitirá incluir un archivo con una estructura marcada.

RFW01.02 Eliminar dicho archivo.

RFW01.03 La generación de un documento de tipo offline y online (secuencial/adaptativa).

RFW02 Se permitirá gestionar asignaturas: añadir, modificar y eliminar.

RFW03 La asociación de asignaturas y documentos.

RFW04 La asociación de asignaturas y alumnos.

■ Alumno

RFW05 Será posible ver los documentos añadidos por el docente, donde sólo podrá ver aquellos a cuyas asignaturas pertenezca.

RFW06 Se podrá descargar las presentaciones offline, y ejecutarlas en un navegador Web, sin conexión a Internet.

RFW07 Y también podrá visualizar las presentaciones online, conectándose al repositorio.

- **Administrador**

RFW08 Será posible gestionar usuarios: añadir, modificar y eliminar.

Generación y visualización de documentos

La generación de documentos formativos debe cumplir lo siguiente:

- **Entrada:** un documento bien formado, con los archivos multimedia definidos, estructuras necesarias para su construcción y estilos a aplicar en el documento resultado.
- **Salida:** una presentación interactiva, constituida por pasos y con reglas de navegación asociadas, cuyos pasos transitarán de manera secuencial/adaptativa.

En cuanto, a la visualización de documentos formativos, los requisitos funcionales son los siguientes:

RFG01 Al visualizar el documento, verá los componentes: narrativo y multimedia, y en el caso de haber componente evaluador, botón *Ver pregunta* para poder visualizarlo.

RFG02 Tendrá que responder las preguntas de tipo test, un valor o rellenar huecos, si quiere visualizar el paso *siguiente*, pero no es necesario para ver el *anterior*.

RFG03 Una vez respondida la pregunta, el documento actuará de manera diferente si la navegación es secuencial o adaptativa:

RFG03.01 Si es secuencial, *siguiente* devolverá el paso definido en el documento, sin importar su respuesta.

RFG03.02 Si es adaptativa, depende de las reglas de navegación, dependiendo del nivel obtenido por él, el documento nos devolverá el paso con nivel más próximo (igual o menor).

RFG04 Si responde de forma incorrecta a la pregunta, el documento mostrará un *feedback*, con información referente a la pregunta, pero no dirá cual era la respuesta correcta.

RFG05 Una vez terminado el documento, vistos todos los pasos, podrá ver el nivel obtenido en cada uno de los conceptos del documento.

3.1.3. Requisitos no funcionales

Los requisitos no funcionales se refieren a aspectos que si bien no intervienen en lo que puede hacer la aplicación, sí son importantes a la hora de utilizarla.

Editor

El editor deberá cumplir los requisitos no funcionales siguientes:

- La *GUI* ha de intentar ser lo más fácil de usar posible.
- El editor debe ser portable a distintos sistemas operativos.

Repositorio

El repositorio deberá cumplir los siguientes requisitos funcionales:

- La interfaz Web ha de intentar ser lo más fácil de usar posible.
- En general, el repositorio debe:
 - Poder ejecutarse en distintos sistemas operativos.
 - Poder usarse por docentes y alumnos.
 - Estar disponible en todo momento y desde cualquier parte del Mundo.

Generación y visualización de documentos

La generación de documentos debe ser lo más rápida posible. Consiguiendo generar todo el código web necesario, crear archivo comprimido (offline) o guardar toda esa información en el servidor (online).

El marco de visualización se adaptará al tamaño de la ventana, permitiendo el máximo área de visualización del contenido del paso. Mientras que en la visualización de estos documentos, los requisitos no funcionales deben ser los siguientes:

- El marco de visualización, de cada paso, se adaptará al tamaño de la ventana, permitiendo una área de visualización máxima de su contenido.
- En la parte inferior, deben mostrarse las flechas de navegación *anterior* y *siguiente*, donde deben ser apreciadas en todo momento.
- Al finalizar el documento, se debe mostrar el conocimiento, de una forma clara y sencilla. Mostrando cada concepto y nivel obtenidos.

3.1.4. Diagrama de Casos de uso

En esta sección, se muestran los distintos casos de uso definidos para el editor, repositorio, y generación y visualización de documentos.

Editor



Figura 3.1: Caso de uso: crear nuevo documento y diagrama

■ Vista: Diagrama de navegación

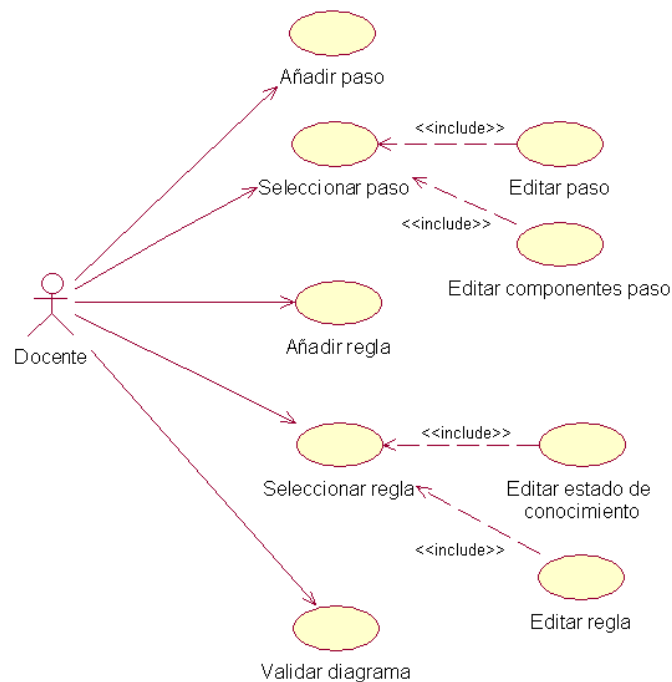


Figura 3.2: Caso de uso: añadir pasos, reglas y validar la navegación

■ Vista: Edición del paso

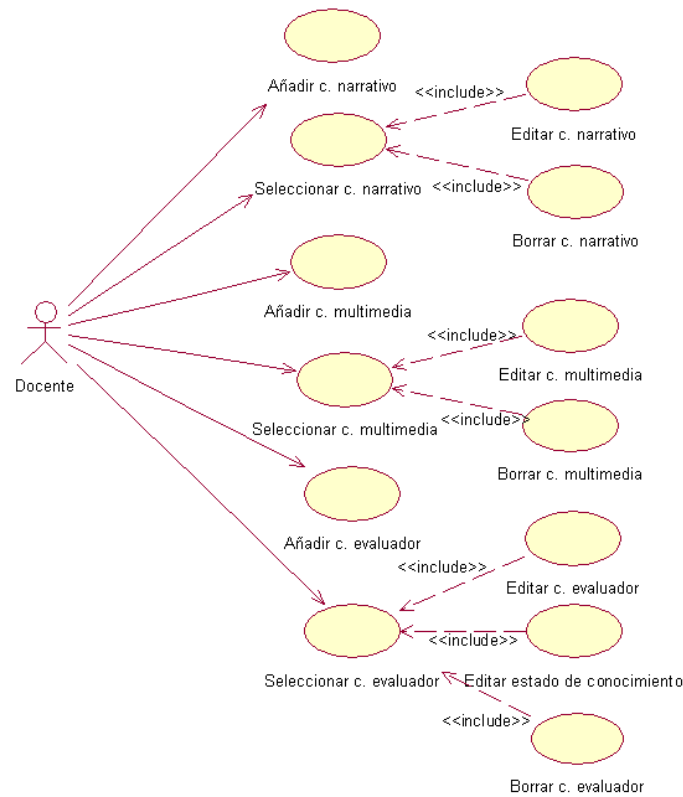


Figura 3.3: Caso de uso: añadir componentes

Repositorio

■ Docente

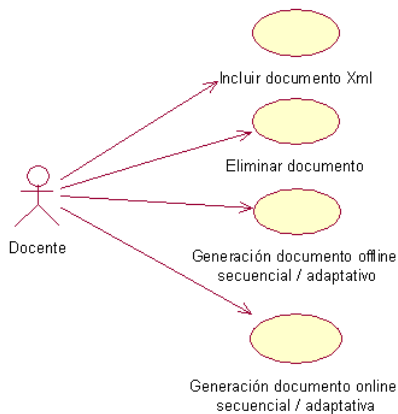


Figura 3.4: Caso de uso: gestión de documentos

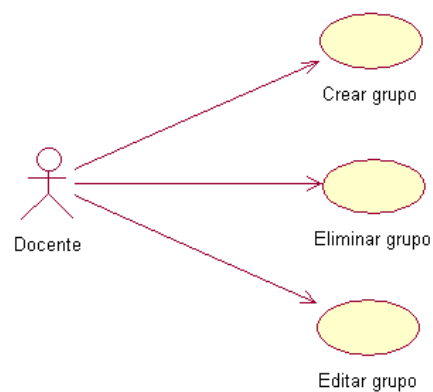


Figura 3.5: Caso de uso: gestión de grupos

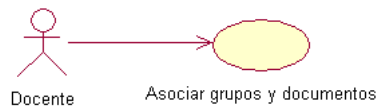


Figura 3.6: Caso de uso: asociación de grupos y documentos

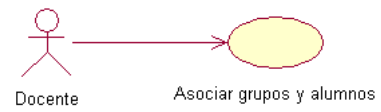


Figura 3.7: Caso de uso: asociación de grupos y alumnos

■ Alumno



Figura 3.8: Caso de uso: descarga de documentos offline

■ Administrador

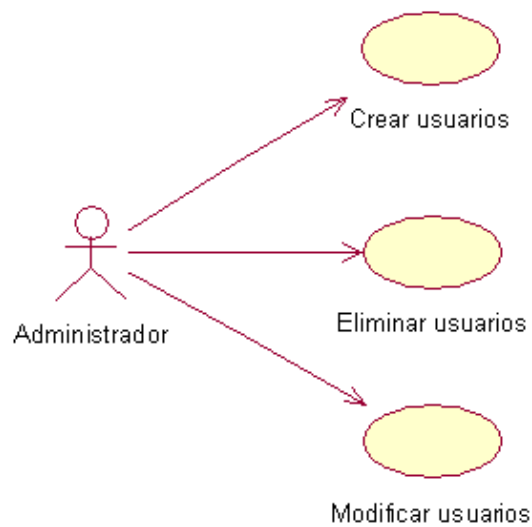


Figura 3.9: Caso de uso: gestión de usuarios

Generación y visualización de documentos

La generación de documentos presenta los siguientes casos de uso:

■ Docente

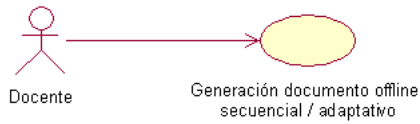


Figura 3.10: Caso de uso: generación de documentos offline

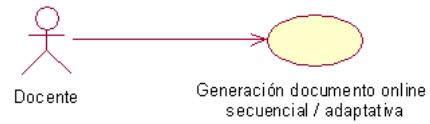


Figura 3.11: Caso de uso: generación de documentos online

Y para la visualización de documentos, se presentan los siguientes:

■ Alumno

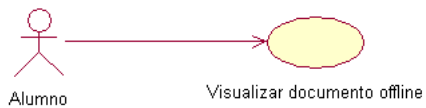


Figura 3.12: Caso de uso: visualización de documentos offline



Figura 3.13: Caso de uso: visualización de documentos online

3.2. ANÁLISIS

En este apartado se va a realizar la especificación de los casos de uso mediante el uso de diagramas de colaboración. En estos diagramas, se pueden reconocer: interfaces (describen interacciones entre el usuario y la aplicación), clases entidad (modelan información persistente o que presenta una larga vida) y clases de control (representan la coordinación, secuenciación y control de otros objetos).

3.2.1. Diagramas de colaboración de los Casos de uso

A continuación se van a mostrar los diagramas de colaboración, para cada caso de uso, detallados con su explicación.

Editor

Caso de uso: Crear documento y diagrama de navegación

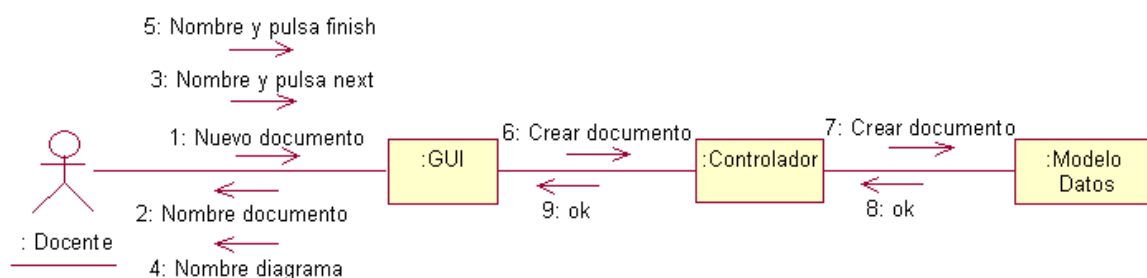


Figura 3.14: Diagrama de colaboración de *Crear documento y diagrama de navegación*

Código: 001	Versión: 1.0
Descripción y desarrollo	En este caso de uso, el docente debe crear un nuevo documento y diagrama de navegación. Previamente, se ha tenido que ejecutar el <i>editor</i> . Después, el docente tendrá que seleccionar en el menú <i>File, New y Documento Diagram</i> . Aparecerá un diálogo donde podrá cambiar el nombre del documento, <i>Next</i> , el nombre del diagrama, y <i>Finish</i> .
Propósito	Crear un nuevo documento.
Actores relacionados	Docente.
Precondiciones	Ejecutar el editor.
Postcondiciones	Crear un nuevo documento y su diagrama de navegación asociado.

■ Vista: Diagrama de navegación

Caso de uso: Añadir paso

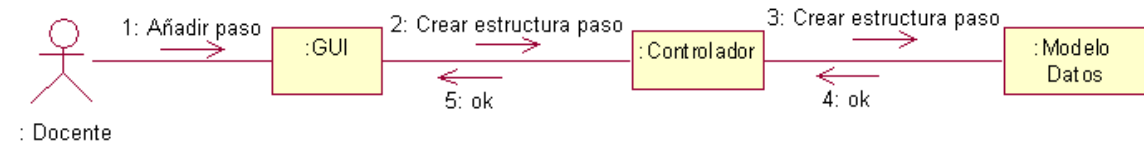


Figura 3.15: Diagrama de colaboración de *Añadir paso*

Código: 002	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el docente debe añadir un paso al diagrama de navegación.</p> <p>Previamente, se han tenido que haber creado un documento y diagrama de navegación. Después, el docente tendrá que seleccionar, de la caja de herramientas, un <i>Paso</i> y hacer <i>click</i> en el diagrama de navegación. Además, de añadir su nombre, en la cabecera de dicho paso.</p>
Propósito	Añadir un paso.
Actores relacionados	Docente.
Precondiciones	Que se haya creado un documento y diagrama de navegación.
Postcondiciones	Añadir un paso a la navegación.

Caso de uso: Editar paso

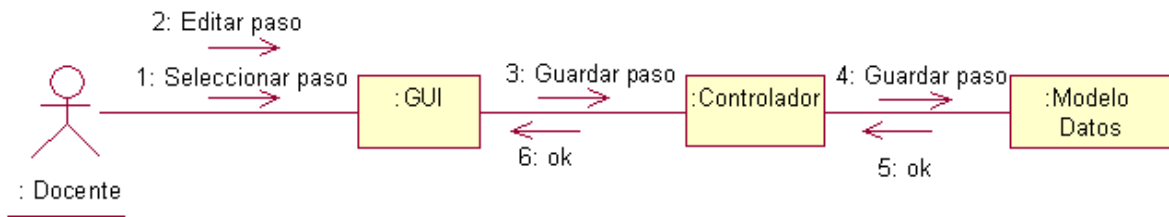
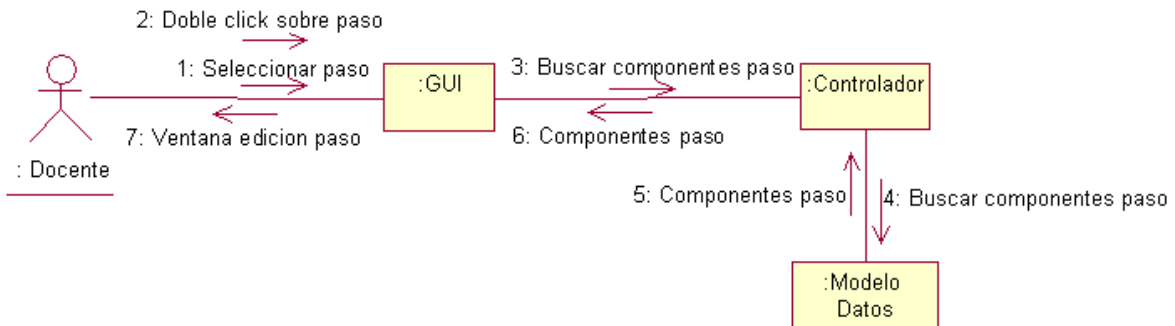


Figura 3.16: Diagrama de colaboración de *Editar paso*

Código: 003	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el docente debe editar un paso del diagrama de navegación.</p> <p>Previamente, se ha tenido que haber incluido un paso al diagrama de navegación. Después, el docente tendrá que seleccionar ese paso, y editar, en la vista de <i>Properties</i>, el <i>Core</i> [Modelo de datos] (Id y Nombre).</p>
Propósito	Editar un paso.
Actores relacionados	Docente.
Precondiciones	Que se haya incluido un paso en el diagrama de navegación.
Postcondiciones	Editar un paso.

Caso de uso: Editar componentes del pasoFigura 3.17: Diagrama de colaboración de *Editar componentes del paso*

Código: 004	Versión: 1.0
Descripción y desarrollo	En este caso de uso, el docente debe editar un paso, añadiéndole componentes a dicho paso. Previamente, se ha tenido que haber incluido un paso al diagrama de navegación. Después, el docente tendrá que seleccionar ese paso, hacer <i>double click</i> sobre él, apareciendo una nueva vista “Vista de edición del paso”. Donde podrá añadir componentes y editarlos.
Propósito	Editar los componentes del paso.
Actores relacionados	Docente.
Precondiciones	Que se haya incluido un paso en el diagrama de navegación.
Postcondiciones	Editar los componentes del paso.

Caso de uso: Añadir regla de navegación

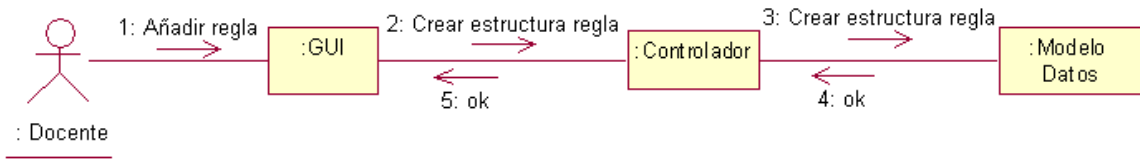


Figura 3.18: Diagrama de colaboración de *Añadir regla de navegación*

Código: 005	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el docente debe añadir una regla al diagrama de navegación.</p> <p>Previamente, se han tenido que haber incluido dos pasos en el diagrama de navegación. Después, el docente tendrá que seleccionar, de la caja de herramientas, una <i>Regla</i> y hacer <i>click</i> en el diagrama de navegación, y unir dos pasos (arco). Además, de añadir su nombre.</p>
Propósito	Añadir una regla al diagrama de navegación.
Actores relacionados	Docente.
Precondiciones	Que se hayan incluido dos pasos en el diagrama de navegación.
Postcondiciones	Añadir una regla al diagrama de navegación que una dos pasos.

Caso de uso: Editar regla

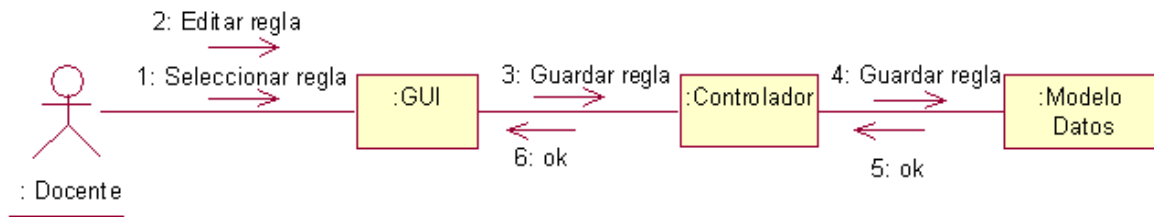


Figura 3.19: Diagrama de colaboración de *Editar regla*

Código: 006	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el docente debe editar una regla al diagrama de navegación.</p> <p>Previamente, se ha tenido que haber incluido una regla en el diagrama de navegación. Después, el docente tendrá que seleccionar esa regla, y podrá editar, en la vista de <i>Properties</i>, el <i>Core</i> [Modelo de datos] (Destino, Id, Nombre y Origen).</p>
Propósito	Editar una regla de navegación.
Actores relacionados	Docente.
Precondiciones	Que se haya incluido una regla en el diagrama de navegación.
Postcondiciones	Editar una regla de navegación.

Caso de uso: Editar estado de conocimiento de la regla

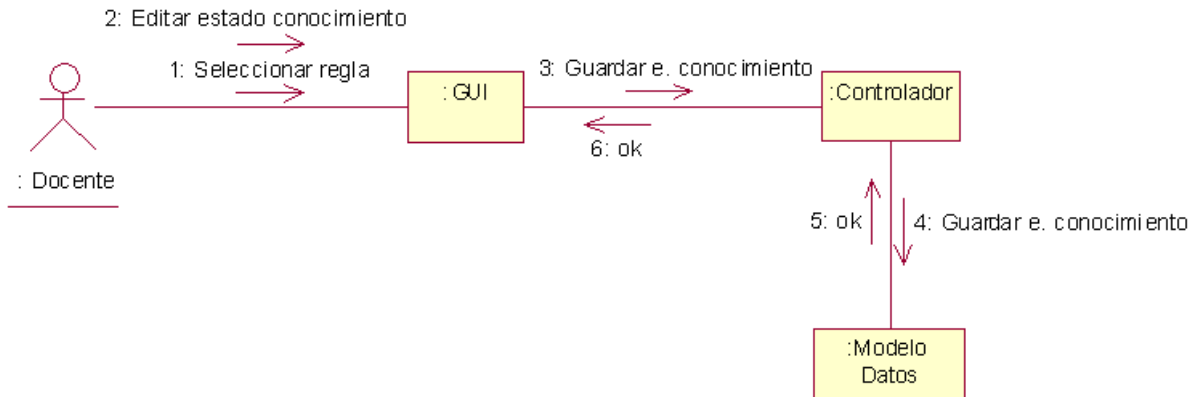
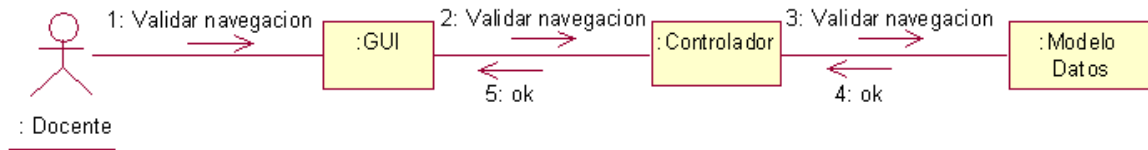


Figura 3.20: Diagrama de colaboración de *Editar estado de conocimiento de la regla*

Código: 007	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el docente debe editar un estado de conocimiento de una regla.</p> <p>Previamente, se ha tenido que haber incluido una regla en el diagrama de navegación. Después, el docente tendrá que seleccionar esa regla, y mostrar el menú contextual (con el botón derecho del ratón), y seleccionar <i>EstadoDeConocimiento</i>. Aparecerá un diálogo donde podrá <i>Añadir</i> y <i>Borrar</i> estados de conocimiento.</p>
Propósito	Editar el estado de conocimiento de una regla de navegación.
Actores relacionados	Docente.
Precondiciones	Que se haya incluido una regla en el diagrama de navegación.
Postcondiciones	Editar el estado de conocimiento de una regla de navegación.

Caso de uso: Validar navegaciónFigura 3.21: Diagrama de colaboración de *Validar navegación*

Código: 008	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el docente debe validar la navegación del diagrama.</p> <p>Previamente, se han tenido que haber creado un documento y diagrama de navegación. Después, el docente tendrá que pulsar sobre el <i>Diagrama de navegación</i>, mostrar el menú contextual (botón derecho), y seleccionar <i>Validar</i>. Aparecerá un diálogo con el listado de errores, que deberá estar vacío para poder generar el documento final.</p>
Propósito	Validar la navegación del documento.
Actores relacionados	Docente.
Precondiciones	Que se haya creado un documento y diagrama de navegación.
Postcondiciones	Validar la navegación del diagrama de navegación.

- **Vista:** Edición del paso

Caso de uso: Añadir componente narrativo

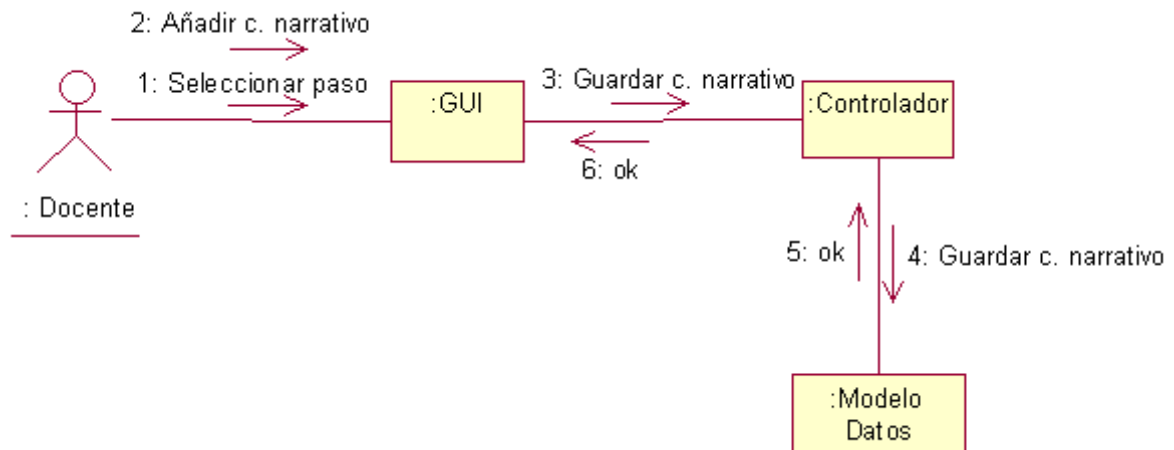


Figura 3.22: Diagrama de colaboración de *Añadir componente narrativo*

Código: 009	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el docente debe añadir un componente narrativo al paso.</p> <p>Previamente, se ha tenido que haber creado un paso, y abrir “Vista de Edición del paso”. Después, el docente tendrá que seleccionar, de la caja de herramientas, <i>Texto</i> y hacer <i>click</i> en el diagrama del paso.</p>
Propósito	Añadir un componente narrativo al paso seleccionado.
Actores relacionados	Docente.
Precondiciones	Que se haya seleccionado un paso.
Postcondiciones	Añadir un componente narrativo al paso seleccionado.

Caso de uso: Editar componente narrativo

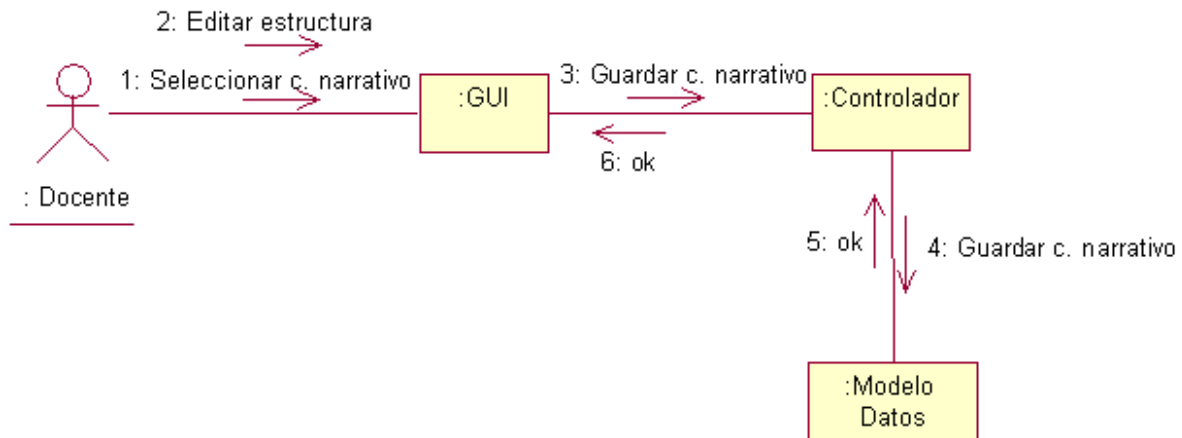


Figura 3.23: Diagrama de colaboración de *Editar componente narrativo*

Código: 010	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el docente debe editar un componente narrativo.</p> <p>Previamente, se ha tenido que haber añadido un componente narrativo. Después, el docente tendrá que seleccionar un componente narrativo, y podrá editar, en la vista de <i>Properties</i>, el <i>Core</i> [Modelo de datos] (Alto, Ancho, Clase CSS, Pos X y Pos Y). Además, podrá mostrar el menú contextual (botón derecho), y seleccionar <i>Modificar</i>, donde cambiar también la <i>Clase CSS</i>, y añadir, modificar y borrar párrafos.</p>
Propósito	Editar un componente narrativo.
Actores relacionados	Docente.
Precondiciones	Que se haya creado un componente narrativo.
Postcondiciones	Editar un componente narrativo.

Caso de uso: Borrar componente narrativo

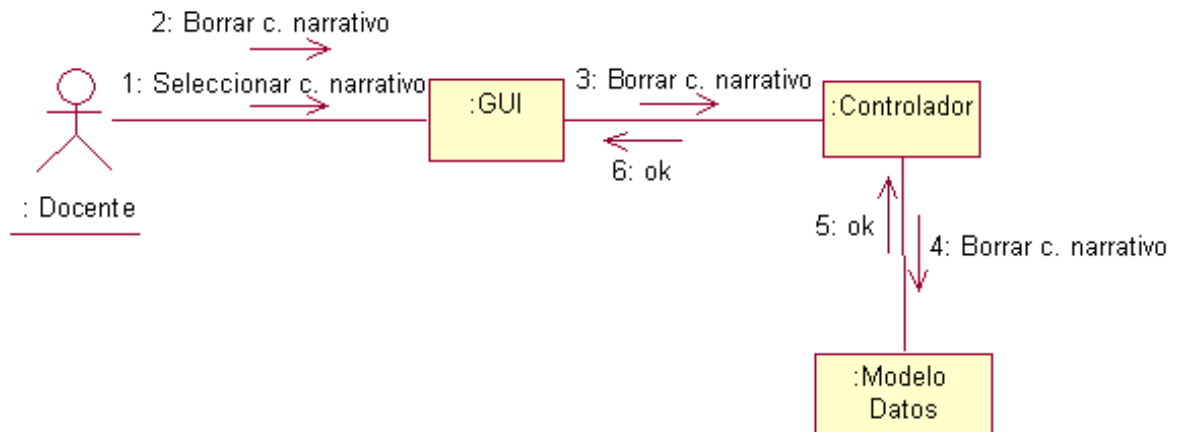


Figura 3.24: Diagrama de colaboración de *Borrar componente narrativo*

Código: 011	Versión: 1.0
Descripción y desarrollo	En este caso de uso, el docente debe borrar un componente narrativo. Previamente, se ha tenido que haber añadido un componente narrativo. Después, el docente tendrá que seleccionar ese componente narrativo, y mostrar el menú contextual (botón derecho), y seleccionar <i>Delete from Model</i> .
Propósito	Borrar un componente narrativo.
Actores relacionados	Docente.
Precondiciones	Que se haya creado un componente narrativo.
Postcondiciones	Borrar un componente narrativo.

Caso de uso: Añadir componente multimedia

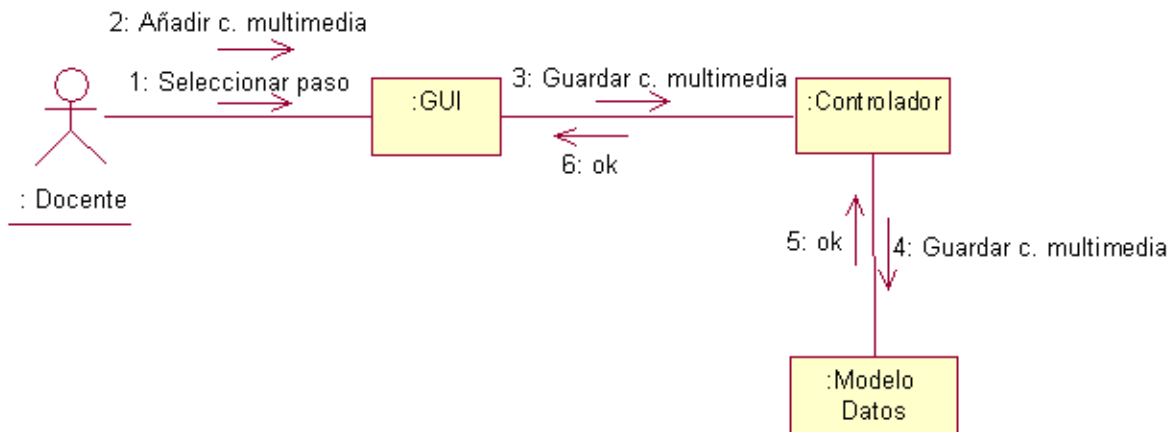


Figura 3.25: Diagrama de colaboración de *Añadir componente multimedia*

Código: 012	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el docente debe añadir un componente multimedia al paso.</p> <p>Previamente, se ha tenido que haber creado un paso, y abrir “Vista de Edición del paso”. Después, el docente tendrá que seleccionar, de la caja de herramientas, <i>Medios</i> y hacer <i>click</i> en el diagrama del paso.</p>
Propósito	Añadir un componente multimedia al paso seleccionado.
Actores relacionados	Docente.
Precondiciones	Que se haya seleccionado un paso.
Postcondiciones	Añadir un componente multimedia al paso seleccionado.

Caso de uso: Editar componente multimedia

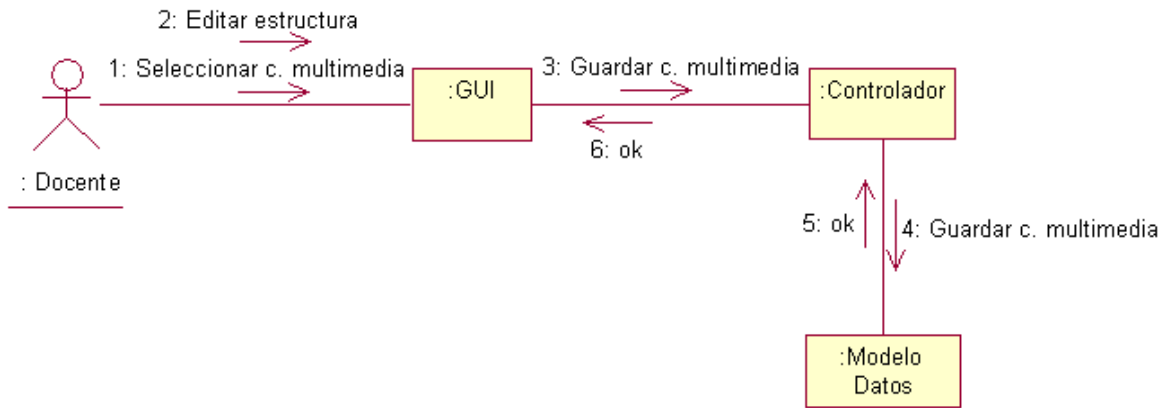


Figura 3.26: Diagrama de colaboración de *Editar componente multimedia*

Código: 013	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el docente debe editar un componente multimedia.</p> <p>Previamente, se ha tenido que haber añadido un componente multimedia. Después, el docente tendrá que seleccionar un componente multimedia, y podrá editar, en la vista de <i>Properties</i>, el <i>Core</i> [Modelo de datos] (Alto, Ancho, Clase CSS, Path Absoluto, Pos X, Pos Y y Tipo). Además, podrá mostrar el menú contextual (botón derecho), y seleccionar <i>Modificar</i>, donde cambiar también la <i>Clase CSS</i>, y seleccionar <i>Tipo</i>, elegir <i>Fichero Local</i>, e introducir los valores de los <i>Parametros</i>.</p>
Propósito	Editar un componente multimedia.
Actores relacionados	Docente.
Precondiciones	Que se haya seleccionado un componente multimedia.
Postcondiciones	Editar un componente multimedia.

Caso de uso: Borrar componente multimedia

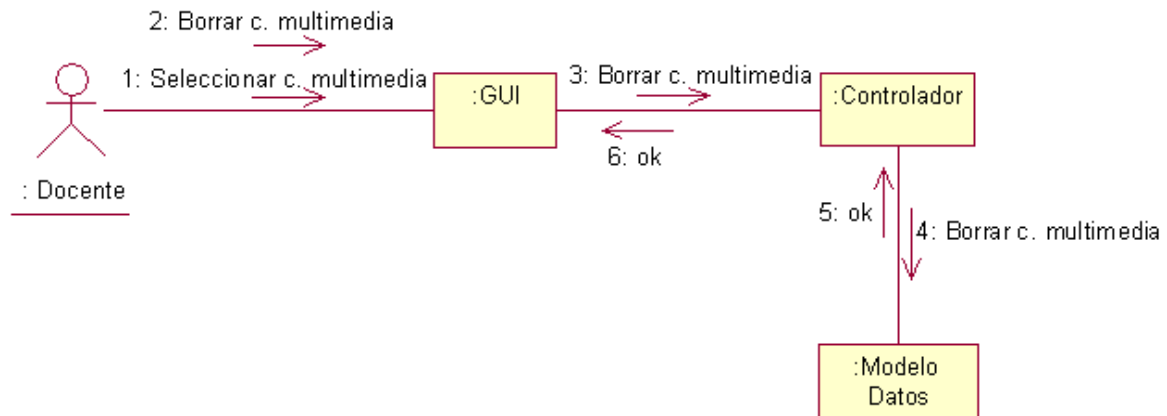


Figura 3.27: Diagrama de colaboración de *Borrar componente multimedia*

Código: 014	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el docente debe borrar un componente multimedia.</p> <p>Previamente, se ha tenido que haber añadido un componente multimedia. Después, el docente tendrá que seleccionar ese componente multimedia, y mostrar el menú contextual (botón derecho), y seleccionar <i>Delete from Model</i>.</p>
Propósito	Borrar un componente multimedia.
Actores relacionados	Docente.
Precondiciones	Que se haya seleccionado un componente multimedia.
Postcondiciones	Borrar un componente multimedia.

Caso de uso: Añadir componente evaluador

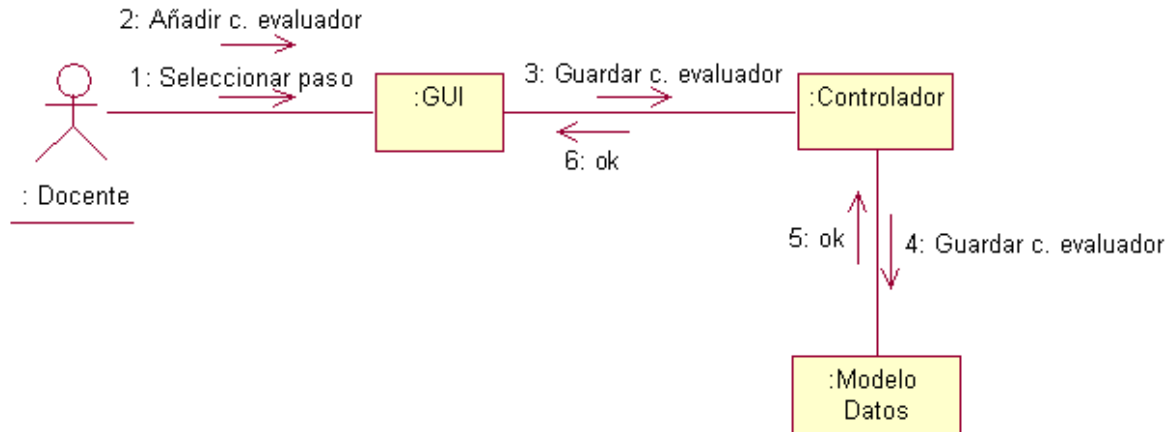
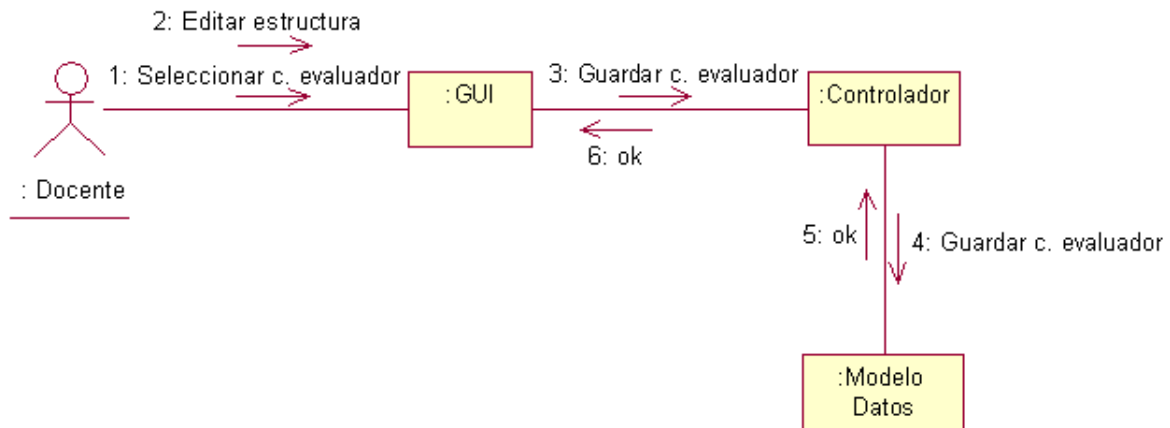


Figura 3.28: Diagrama de colaboración de *Añadir componente evaluador*

Código: 015	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el docente debe añadir un componente evaluador al paso.</p> <p>Previamente, se ha tenido que haber creado un paso, y abrir “Vista de Edición del paso”. Después, el docente tendrá que seleccionar, de la caja de herramientas, <i>Pregunta</i> y hacer <i>click</i> en el diagrama del paso.</p>
Propósito	Añadir un componente evaluador al paso seleccionado.
Actores relacionados	Docente.
Precondiciones	Que se haya seleccionado un paso.
Postcondiciones	Añadir un componente evaluador al paso seleccionado.

Caso de uso: Editar componente evaluadorFigura 3.29: Diagrama de colaboración de *Editar componente evaluador*

Código: 016	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el docente debe editar un componente evaluador.</p> <p>Previamente, se ha tenido que haber añadido un componente evaluador. Después, el docente tendrá que seleccionar un componente evaluador, y podrá editar, en la vista de <i>Properties</i>, el <i>Core</i> [Modelo de datos] (Alto, Ancho, Clase CSS, Path Absoluto, Pos X y Pos Y). Además, podrá mostrar el menú contextual (botón derecho), y seleccionar <i>Modificar</i>, donde cambiar también la <i>Clase CSS</i>, añadir, modificar y borrar párrafos. Una vez elegido el <i>Tipo</i> de pregunta, podrá añadir respuestas de tipo <i>Test</i>, <i>Un Valor</i> y <i>Rellenar Huecos</i>.</p> <p>Si Test, al pulsar <i>Añadir</i>, aparecerá un nuevo diálogo donde podrá cambiar también la <i>Clase CSS</i>, elegir <i>Respuesta Correcta</i>, <i>Añadir Parrafo</i> (con un nuevo diálogo para añadir párrafos a las respuestas) y escribir <i>FeedBack</i>, donde aparecerá otro diálogo donde añadir párrafos.</p> <p>Si Un Valor, escribir <i>Valor Correcto</i> y añadir <i>FeedBack</i>, donde aparecerá otro diálogo donde incluir párrafos.</p>

Descripción y desarrollo	Si Rellenar Huecos , al pulsar <i>Añadir</i> , aparecerá un nuevo diálogo donde podrá cambiar también la <i>Clase CSS</i> , <i>Añadir Texto</i> , apareciendo de nuevo un diálogo donde escribir <i>Texto</i> y elegir <i>Css</i> . Además, podrá <i>Modificar</i> , <i>Borrar</i> y, elegir el texto para convertir en un hueco, al pulsar <i>Valor</i> .
Propósito	Editar un componente evaluador.
Actores relacionados	Docente.
Precondiciones	Que se haya seleccionado un componente evaluador.
Postcondiciones	Editar un componente evaluador.

Caso de uso: Editar estado de conocimiento del componente evaluador

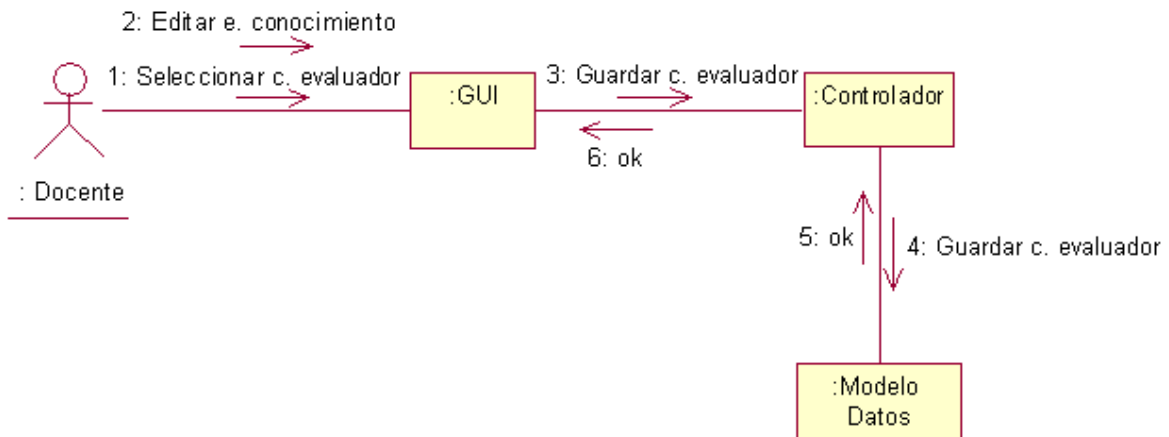


Figura 3.30: Diagrama de colaboración de *Editar estado de conocimiento del componente evaluador*

Código: 017	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el docente debe editar el estado de conocimiento de un componente evaluador.</p> <p>Previamente, se ha tenido que haber añadido un componente evaluador, con una pregunta. Después, el docente tendrá que seleccionar un componente evaluador, y mostrar el menú contextual (botón derecho), y seleccionar <i>EstadoDeConocimiento</i>, donde podrá <i>Añadir</i> y <i>Borrar</i> estados de conocimiento al componente evaluador.</p>
Propósito	Editar el estado de conocimiento de un componente evaluador.
Actores relacionados	Docente.
Precondiciones	Que se haya seleccionado un componente evaluador y tenga una pregunta añadida.
Postcondiciones	Editar el estado de conocimiento de un componente evaluador.

Caso de uso: Borrar componente evaluador

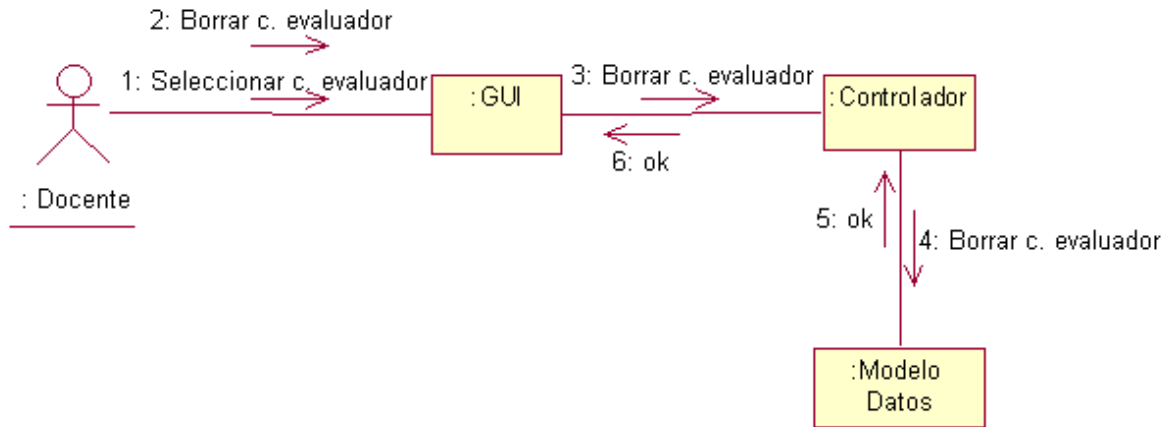


Figura 3.31: Diagrama de colaboración de *Borrar componente evaluador*

Código: 018	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el docente debe borrar un componente evaluador.</p> <p>Previamente, se ha tenido que haber añadido un componente evaluador. Después, el docente tendrá que seleccionar ese componente evaluador, y mostrar el menú contextual (botón derecho), y seleccionar <i>Delete from Model</i>.</p>
Propósito	Borrar un componente evaluador.
Actores relacionados	Docente.
Precondiciones	Que se haya seleccionado un componente evaluador.
Postcondiciones	Borrar un componente evaluador.

Repositorio

■ Docente

Caso de uso: Incluir archivo

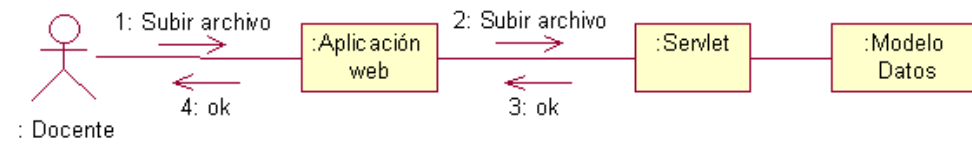


Figura 3.32: Diagrama de colaboración de *Incluir archivo*

Código: 019	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el docente se encargará de incluir el archivo en el servidor.</p> <p>Previamente, el docente tendrá que acceder al <i>repositorio</i> e identificarse mediante su usuario y contraseña. Después, el docente tendrá que “Seleccionar archivo” de su equipo, elegir un archivo comprimido .zip (con el documento Xml, archivos multimedia y estructuras), y “Subir documento”. El sistema, comprueba que todo sea correcto y lo vuelca al servidor Web. Si hay algún problema, mostrará un mensaje de error.</p>
Propósito	Subir documento al Servidor Web.
Actores relacionados	Docente.
Precondiciones	Que el docente se haya validado en el repositorio y que haya algún documento en la tabla.
Postcondiciones	Añadir un nuevo archivo al Servidor Web.

Caso de uso: Eliminar archivo

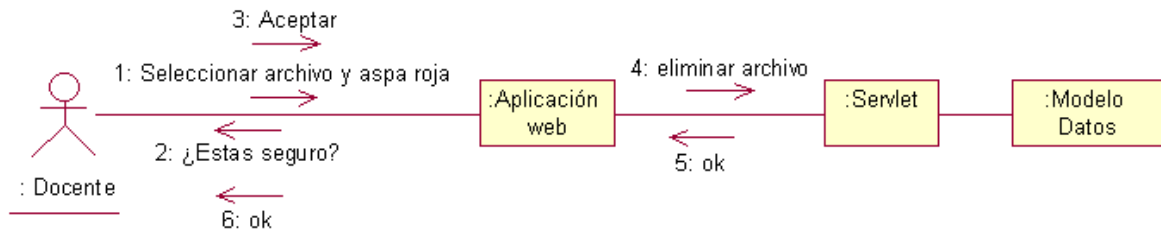
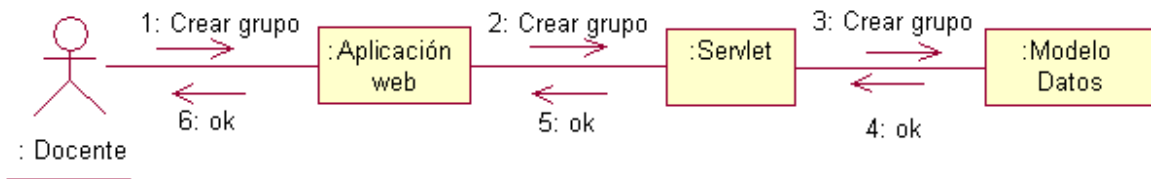
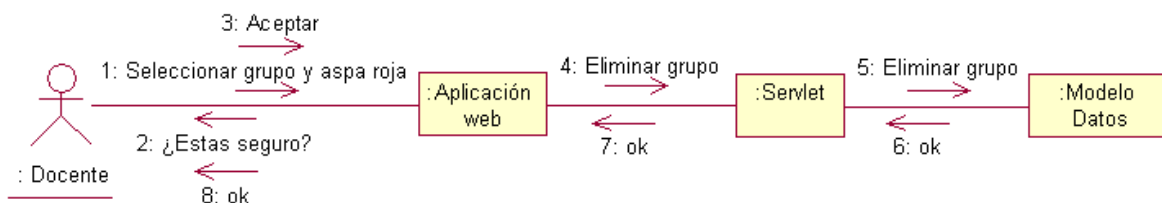


Figura 3.33: Diagrama de colaboración de *Eliminar archivo*

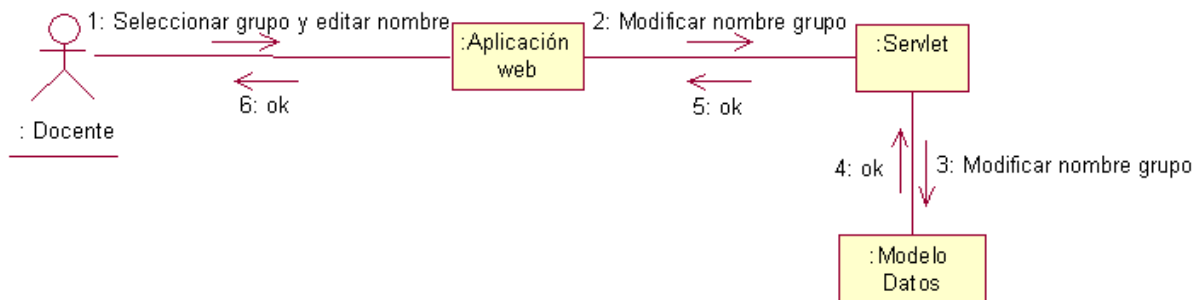
Código: 020	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el docente se encargará de eliminar el archivo Xml del servidor.</p> <p>Previamente, el docente tendrá que acceder al <i>repositorio</i> e identificarse mediante su usuario y contraseña. Después, el docente podrá buscar el archivo, ordenando mediante las flechas que aparecen en la tabla. Cuando lo haya encontrado, pulsará el botón con forma de aspa roja. El sistema, mostrará un mensaje “¿Está seguro que desea eliminar este documento?”, pulsará “Aceptar” y el sistema eliminará el documento del servidor, si pulsa “Cancelar” se abortará el borrado del documento.</p>
Propósito	Eliminar documento del Servidor Web.
Actores relacionados	Docente.
Precondiciones	Que el docente se haya validado en el repositorio y que haya algún documento en la tabla.
Postcondiciones	Eliminar el archivo del Servidor Web.

Caso de uso: Crear GrupoFigura 3.34: Diagrama de colaboración de *Crear Grupo*

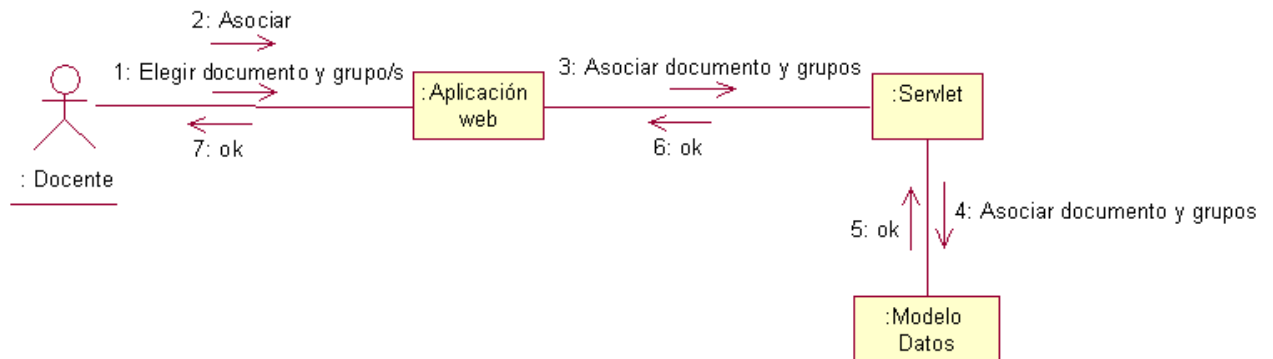
Código: 021	Versión: 1.0
Descripción y desarrollo	En este caso de uso, el docente podrá crear grupos. Previamente, el docente tendrá que acceder al <i>repositorio</i> e identificarse mediante su usuario y contraseña. Después, el docente podrá escribir el <i>Nombre del grupo</i> y “Crear grupo”, el servidor añadirá este grupo en base de datos y actualizará la tabla de pantalla.
Propósito	Crear grupos en la base de datos y en el repositorio.
Actores relacionados	Docente.
Precondiciones	Que el docente se haya validado en el repositorio.
Postcondiciones	Crear un grupo.

Caso de uso: Eliminar GrupoFigura 3.35: Diagrama de colaboración de *Eliminar Grupo*

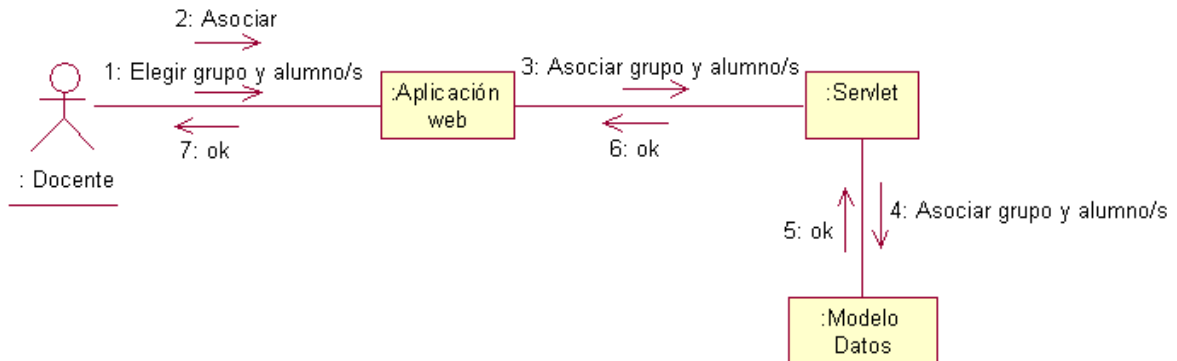
Código: 022	Versión: 1.0
Descripción y desarrollo	En este caso de uso, el docente podrá eliminar grupos. Previamente, el docente tendrá que acceder al <i>repositorio</i> e identificarse mediante su usuario y contraseña. Después, el docente podrá buscar el grupo, y una vez encontrado, podrá pulsar el aspa roja. El sistema, mostrará un mensaje “¿Está seguro que desea eliminar este grupo?”, pulsará “Aceptar” y el sistema eliminará el grupo de la tabla y de la base de datos, si pulsa “Cancelar” se abortará el borrado del grupo.
Propósito	Eliminar grupos de base de datos y del repositorio.
Actores relacionados	Docente.
Precondiciones	Que el docente se haya validado en el repositorio y que haya algún grupo en la tabla.
Postcondiciones	Eliminar un grupo.

Caso de uso: Editar GrupoFigura 3.36: Diagrama de colaboración de *Editar Grupo*

Código: 023	Versión: 1.0
Descripción y desarrollo	En este caso de uso, el docente podrá editar grupos. Previamente, el docente tendrá que acceder al <i>repositorio</i> e identificarse mediante su usuario y contraseña. Después, el docente podrá buscar el grupo, y una vez encontrado, podrá pulsar el icono junto al aspa roja (lápiz sobre un cuaderno). El sistema, mostrará el grupo elegido en la parte superior de pantalla (Nombre del grupo) y podrá ser modificado su nombre, al pulsar "Editar". El sistema modificará el nombre en la tabla y en base de datos.
Propósito	Editar el nombre de los grupos de base de datos y del repositorio.
Actores relacionados	Docente.
Precondiciones	Que el docente se haya validado en el repositorio y que haya algún grupo en la tabla.
Postcondiciones	Editar un grupo.

Caso de uso: Asociar grupos y documentosFigura 3.37: Diagrama de colaboración de *Asociar grupos y documentos*

Código: 024	Versión: 1.0
Descripción y desarrollo	En este caso de uso, el docente podrá asociar grupos y documentos. Previamente, el docente tendrá que acceder al <i>repositorio</i> e identificarse mediante su usuario y contraseña. Después, el docente podrá elegir un documento y asociar un o más grupos, y pulsar “Asociar”. El sistema actualizará la tabla de pantalla y la base de datos.
Propósito	Asociar grupos y documentos en base de datos y en el repositorio.
Actores relacionados	Docente.
Precondiciones	Que el docente se haya validado en el repositorio y que haya algún documento y algún grupo creados.
Postcondiciones	Asociar grupos a documentos.

Caso de uso: Asociar grupos y alumnosFigura 3.38: Diagrama de colaboración de *Asociar grupos y alumnos*

Código: 025	Versión: 1.0
Descripción y desarrollo	En este caso de uso, el docente podrá asociar grupos y alumnos. Previamente, el docente tendrá que acceder al <i>repositorio</i> e identificarse mediante su usuario y contraseña. Después, el docente podrá elegir un grupo y asociar un o más alumnos, y pulsar “Asociar”. El sistema actualizará la tabla de pantalla y la base de datos.
Propósito	Asociar grupos y documentos en base de datos y en el repositorio.
Actores relacionados	Docente.
Precondiciones	Que el docente se haya validado en el repositorio y que haya algún grupo y algún alumno creados.
Postcondiciones	Asociar grupos a alumnos.

- Alumno

Caso de uso: Descargar documento offline

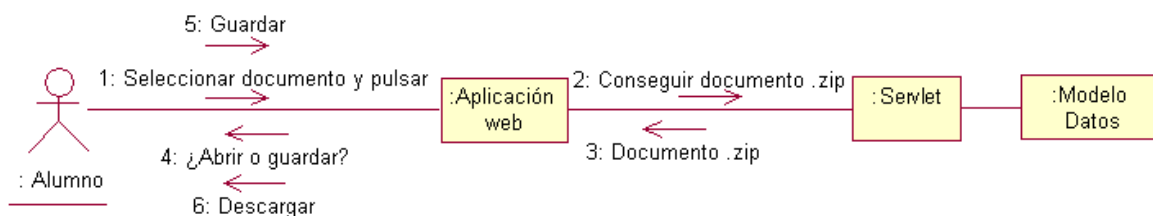


Figura 3.39: Diagrama de colaboración de *Descargar documento offline*

Código: 026	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el alumno podrá descargar documentos offline.</p> <p>Previamente, el alumno tendrá que acceder al <i>repositorio</i> e identificarse mediante su usuario y contraseña. Después, el alumno podrá buscar el documento del grupo que quiere descargar, y una vez encontrado, podrá pulsar en el documento de la columna “Path Offline”. El sistema comenzará la descarga del documento en su directorio de descargas habitual.</p>
Propósito	Descargar el documento offline del repositorio.
Actores relacionados	Alumno.
Precondiciones	Que el alumno se haya validado en el repositorio y que haya algún documento en la tabla.
Postcondiciones	Descargar el documento offline seleccionado.

■ Administrador

Caso de uso: Crear usuarios

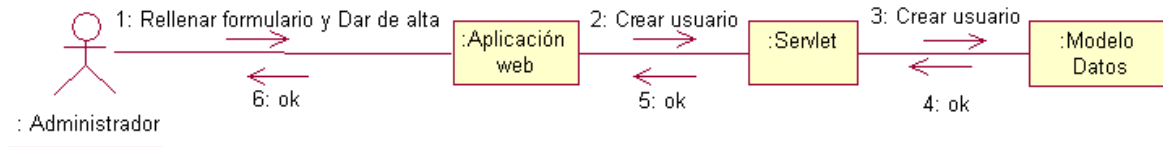


Figura 3.40: Diagrama de colaboración de *Crear usuarios*

Código: 027	Versión: 1.0
Descripción y desarrollo	En este caso de uso, el alumno podrá dar de alta usuarios. Previamente, el administrador tendrá que acceder al <i>repositorio</i> e identificarse mediante su usuario y contraseña. Después, el administrador podrá crear usuarios, introduciendo en el formulario: Usuario, correo electrónico, Contraseña y perfil, y pulsar “Dar de alta”. El sistema actualizará la tabla de pantalla y en la base de datos.
Propósito	Crear usuarios en base de datos y en el repositorio.
Actores relacionados	Administrador.
Precondiciones	Que el administrador se haya validado en el repositorio.
Postcondiciones	Crear usuarios.

Caso de uso: Eliminar usuarios

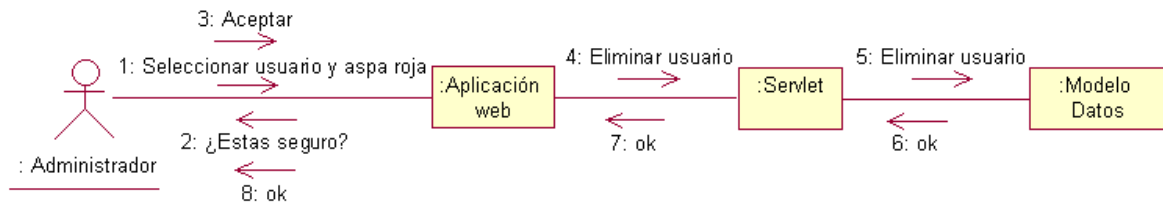


Figura 3.41: Diagrama de colaboración de *Crear usuarios*

Código: 028	Versión: 1.0
Descripción y desarrollo	En este caso de uso, el alumno podrá eliminar usuarios. Previamente, el administrador tendrá que acceder al <i>repositorio</i> e identificarse mediante su usuario y contraseña. Después, el administrador podrá buscar el usuario que quiere eliminar, y una vez encontrado, podrá pulsar el aspa roja. El sistema, mostrará un mensaje “¿Está seguro que desea eliminar este usuario?”, pulsará “Aceptar” y el sistema eliminará el usuario de la tabla y de la base de datos, si pulsa “Cancelar” se abortará el borrado del usuario.
Propósito	Eliminar usuarios de base de datos y del repositorio.
Actores relacionados	Administrador.
Precondiciones	Que el administrador se haya validado en el repositorio y que haya algún usuario en la tabla.
Postcondiciones	Eliminar usuarios.

Caso de uso: Modificar usuarios

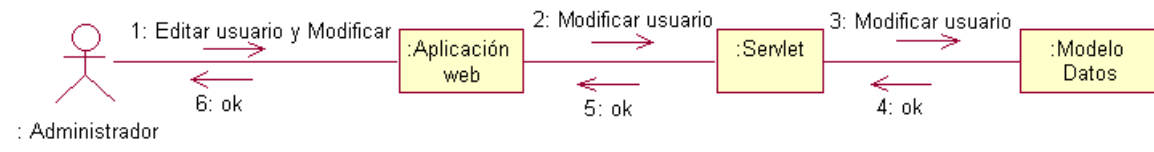


Figura 3.42: Diagrama de colaboración de *Modificar usuarios*

Código: 029	Versión: 1.0
Descripción y desarrollo	En este caso de uso, el alumno podrá modificar usuarios. Previamente, el administrador tendrá que acceder al <i>repositorio</i> e identificarse mediante su usuario y contraseña. Después, el administrador podrá buscar el usuario, y una vez encontrado, podrá pulsar el icono junto al aspa roja (lápiz sobre un cuaderno). El sistema, mostrará el usuario elegido en la parte superior de pantalla y podrá ser modificado su Usuario, correo y perfil solamente, al pulsar "Modificar". El sistema modificará los datos del usuario en la tabla y en la base de datos.
Propósito	Modificar usuarios de base de datos y del repositorio.
Actores relacionados	Administrador.
Precondiciones	Que el administrador se haya validado en el repositorio y que haya algún usuario en la tabla.
Postcondiciones	Modificar usuarios.

Generación y visualización de documentos

Diagramas de colaboración para la generación de documentos:

▪ Docente

Caso de uso: Generar Offline secuencial o adaptativa

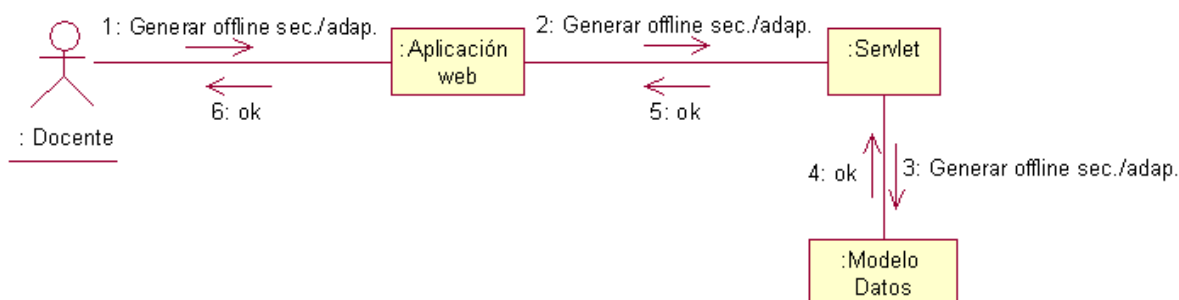
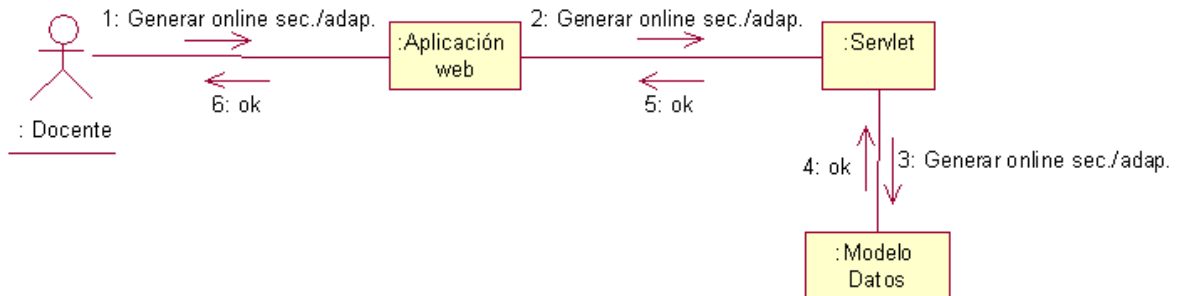


Figura 3.43: Diagrama de colaboración de *Generar Offline secuencial o adaptativa*

Código: 030	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el docente podrá generar documento offline secuencial o adaptativo, dependiendo del pulsado.</p> <p>Previamente, el docente tendrá que acceder al <i>repositorio</i> e identificarse mediante su usuario y contraseña. Después, el docente podrá buscar el archivo, y una vez encontrado, podrá pulsar “Generar Offline secuencial” o “Generar Offline adaptativa”. El sistema, comenzará el proceso de generación y al terminar mostrará un mensaje “La generación se ha realizado correctamente.” en caso de terminar correctamente, y sino “La generación no se ha realizado correctamente, vuelva a intentarlo.” si algo ha fallado.</p>
Propósito	Generar documentos offline con navegación secuencial o adaptativa.
Actores relacionados	Docente.
Precondiciones	Que el docente se haya validado en el repositorio y que haya algún documento en la tabla.
Postcondiciones	Generar el documento offline seleccionado.

Caso de uso: Generar Online secuencial o adaptativaFigura 3.44: Diagrama de colaboración de *Generar Online secuencial o adaptativa*

Código: 031	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el docente podrá generar documento online secuencial o adaptativo, dependiendo del pulsado.</p> <p>Previamente, el docente tendrá que acceder al <i>repositorio</i> e identificarse mediante su usuario y contraseña. Después, el docente podrá buscar el archivo, y una vez encontrado, podrá pulsar “Generar Online secuencial” o “Generar Online adaptativo”. El sistema, comenzará el proceso de generación y al terminar mostrará un mensaje “La generación se ha realizado correctamente.” en caso de terminar correctamente, y sino “La generación no se ha realizado correctamente, vuelva a intentarlo.” si algo ha fallado.</p>
Propósito	Generar documentos online con navegación secuencial o adaptativa.
Actores relacionados	Docente.
Precondiciones	Que el docente se haya validado en el repositorio y que haya algún documento en la tabla.
Postcondiciones	Generar el documento online seleccionado.

Diagramas de colaboración para la visualización de documentos formativos:

■ **Alumno**

Caso de uso: Visualizar documento online

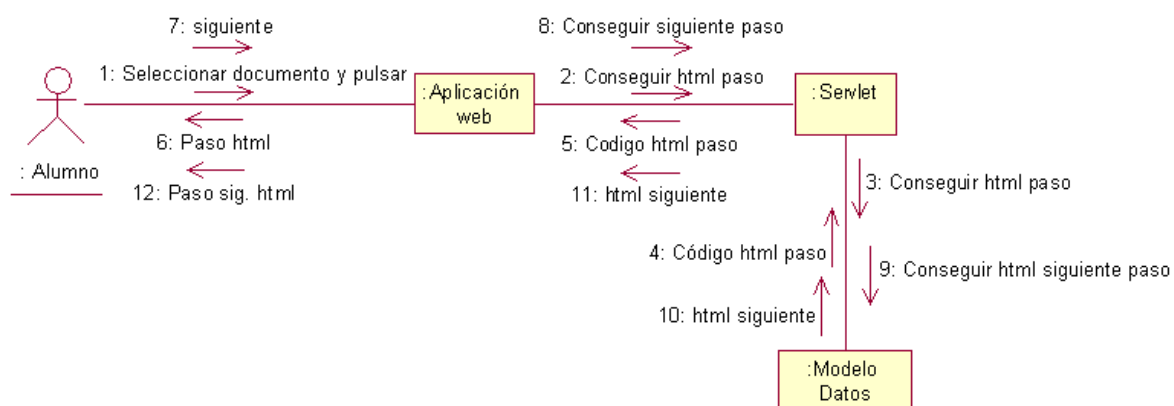


Figura 3.45: Diagrama de colaboración de *Visualizar documento online*

Código: 032	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el alumno podrá visualizar documentos online.</p> <p>Previamente, el alumno tendrá que acceder al <i>repositorio</i> e identificarse mediante su usuario y contraseña. Después, el alumno podrá buscar el documento del grupo que quiere visualizar, y una vez encontrado, podrá pulsar en el documento de la columna “Path Online”. El sistema abrirá una nueva ventana en el navegador, y comenzará la visualización online. Tras la visualización del documento, el sistema mostrará el resultado alcanzado del alumno.</p>
Propósito	Visualizar un documento en online.
Actores relacionados	Alumno.
Precondiciones	Que el alumno se haya validado en el repositorio y que haya algún documento en la tabla.
Postcondiciones	Visualizar el documento online seleccionado.

Caso de uso: Visualizar documento offline

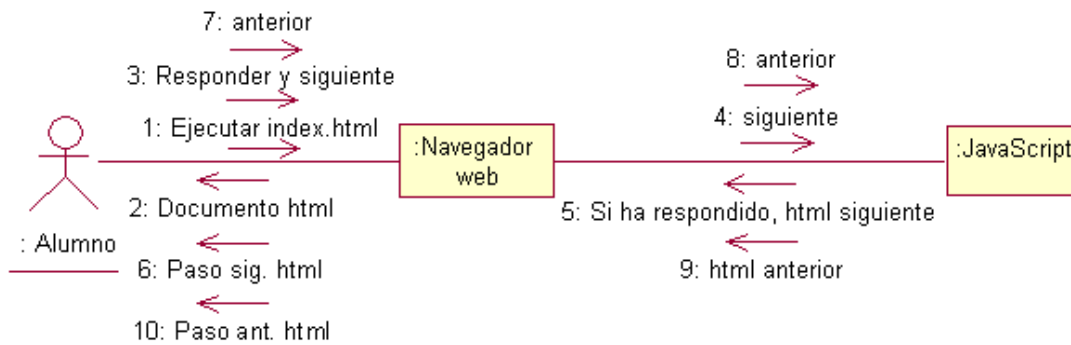


Figura 3.46: Diagrama de colaboración de *Visualizar documento offline*

Código: 033	Versión: 1.0
Descripción y desarrollo	<p>En este caso de uso, el alumno podrá visualizar documentos offline.</p> <p>Previamente, el alumno tendrá que buscar en el directorio de descargas habitual el archivo .zip descargado, descomprimirlo y pulsar el archivo “index.html” que lanzará el navegador que tengamos por defecto.</p>
Propósito	Visualizar un documento en offline.
Actores relacionados	Alumno.
Precondiciones	Que el alumno se haya descargado algún documento.
Postcondiciones	Visualizar el documento offline seleccionado.

3.3. DISEÑO

Para el diseño del sistema se ha hecho uso del lenguaje UML.

3.3.1. Diagrama de clases: Modelo de Datos del documento

El modelo de datos diseñado, para el documento, consta de 22 clases: Documento, Paso, Regla, Componente, ComponenteEvaluador, ComponenteMultimedia, ComponenteNarrativo, Parametro, Enunciado, TextoCss, Pista, Pregunta, PreguntaTest, PreguntaValor, RespuestaTest, RespuestaValor, Feedback, Conocimiento, EstadoDeConocimiento, ConocimientoException, ErrorNavegacion y CadenaErrorNavegacion.

Documento

En la figura 3.47 se muestra la clase *Documento* que contiene la definición del documento y se encarga de devolver los pasos, reglas, ficheros multimedia y el titulo, y validar la navegación.

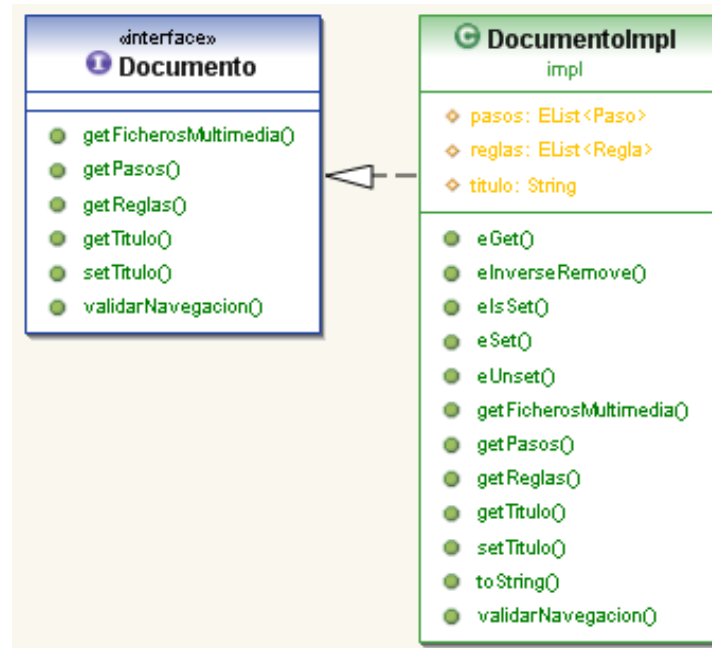


Figura 3.47: Diagrama de clases de *Documento*

Paso y Regla

En la figura 3.48 se muestra la clase *Paso* que contiene la definición del paso y se encarga de devolver los componentes multimedia, narrativo y evaluador, y validar la navegación.

En la figura 3.49 se muestra la clase *Regla* se encarga de modelar las transiciones entre pasos, devolver el estado de conocimiento y validar la navegación.

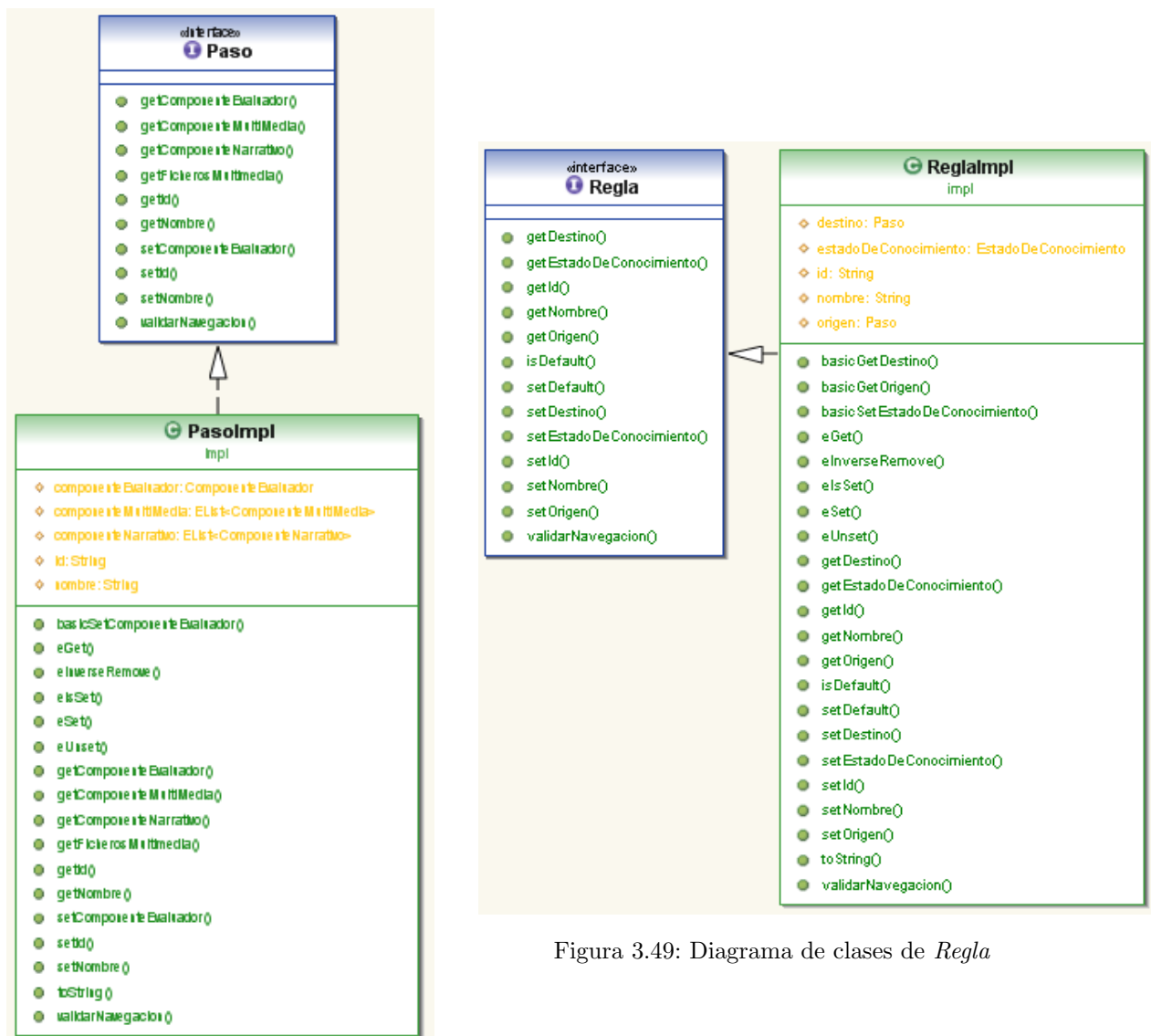


Figura 3.49: Diagrama de clases de *Regla*

Figura 3.48: Diagrama de clases de *Paso*

Componentes

En la figura 3.50 se muestran los diferentes componentes que puede contener un paso: *Componente narrativo*, para el texto, *Componente multimedia*, para mostrar imágenes, sonido, video u otros, y *Componente evaluador*, para las preguntas.

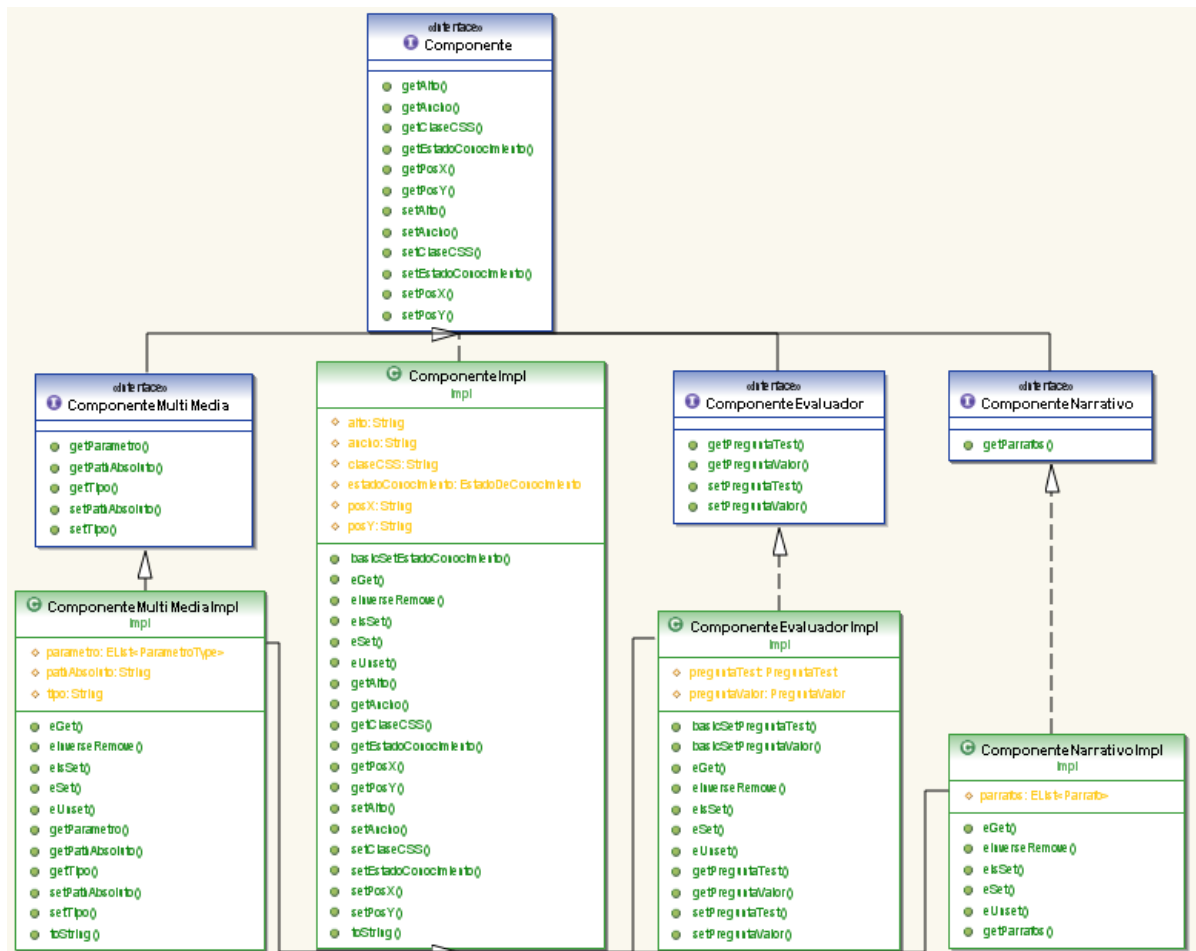


Figura 3.50: Diagrama de clases de *Componentes*: narrativo, multimedia y evaluador

Parametro

En la figura 3.51 se muestra la clase *Parametro* que se encarga de devolver el nombre y valor que lo conforman. Esta clase se utiliza en los componentes multimedia para especificar cada parámetro de las estructuras HTML.

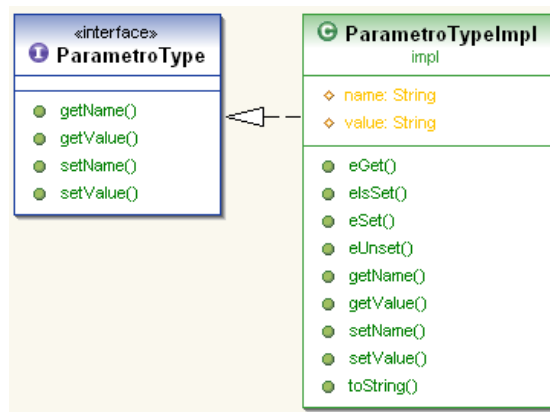


Figura 3.51: Diagrama de clases de *Parametro*

Enunciado y Parrafo

En la figura 3.52 se muestra la clase *Parrafo* que se encarga de obtener los textos y clases de estilo que contiene. En la figura 3.53 se muestra la clase *Parrafo* que se encarga de devolver los párrafos que contiene.

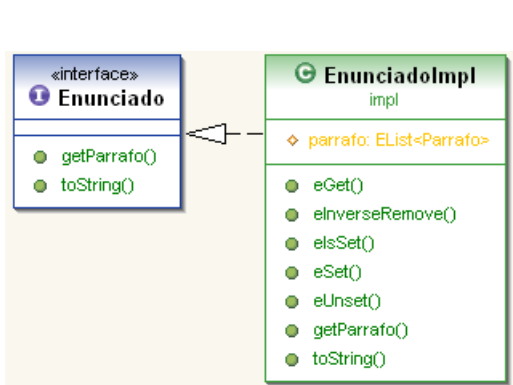


Figura 3.52: Diagrama de clases de *Enunciado*

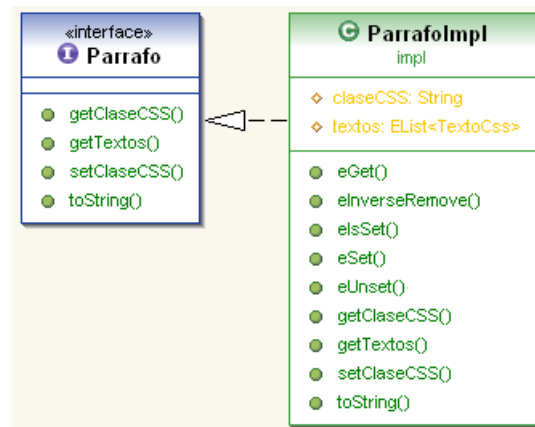


Figura 3.53: Diagrama de clases de *Parrafo*

TextoCss

En la figura 3.54 se muestra la clase *TextoCss*, que se encarga de devolver el texto y las clases de estilo, que utiliza la clase *Parrafo* para constituir los párrafos.

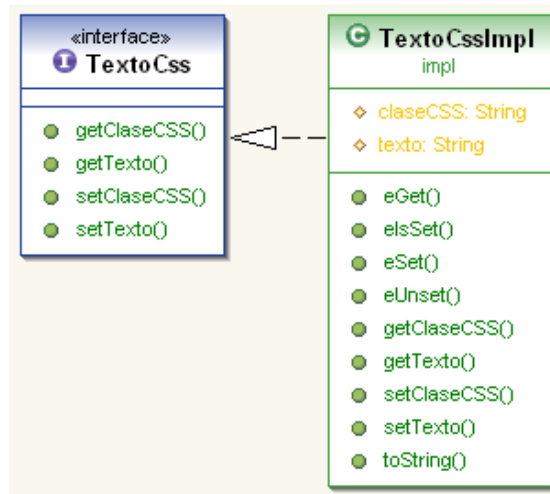


Figura 3.54: Diagrama de clases de *TextoCss*

Pista

En la figura 3.55 se muestra la clase *Pista*, que se encarga de devolver una pista para ayudar al alumno a responder la pregunta planteada.

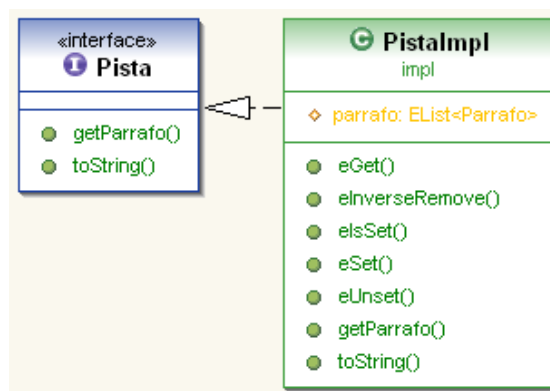


Figura 3.55: Diagrama de clases de *Pista*

Preguntas

En la figura 3.56 se muestran las diferentes preguntas que puede contener un componente evaluador: *Pregunta test*, para las preguntas de tipo test de una respuesta o multiopción, y *Pregunta valor*, para las preguntas de un valor o rellenar huecos.

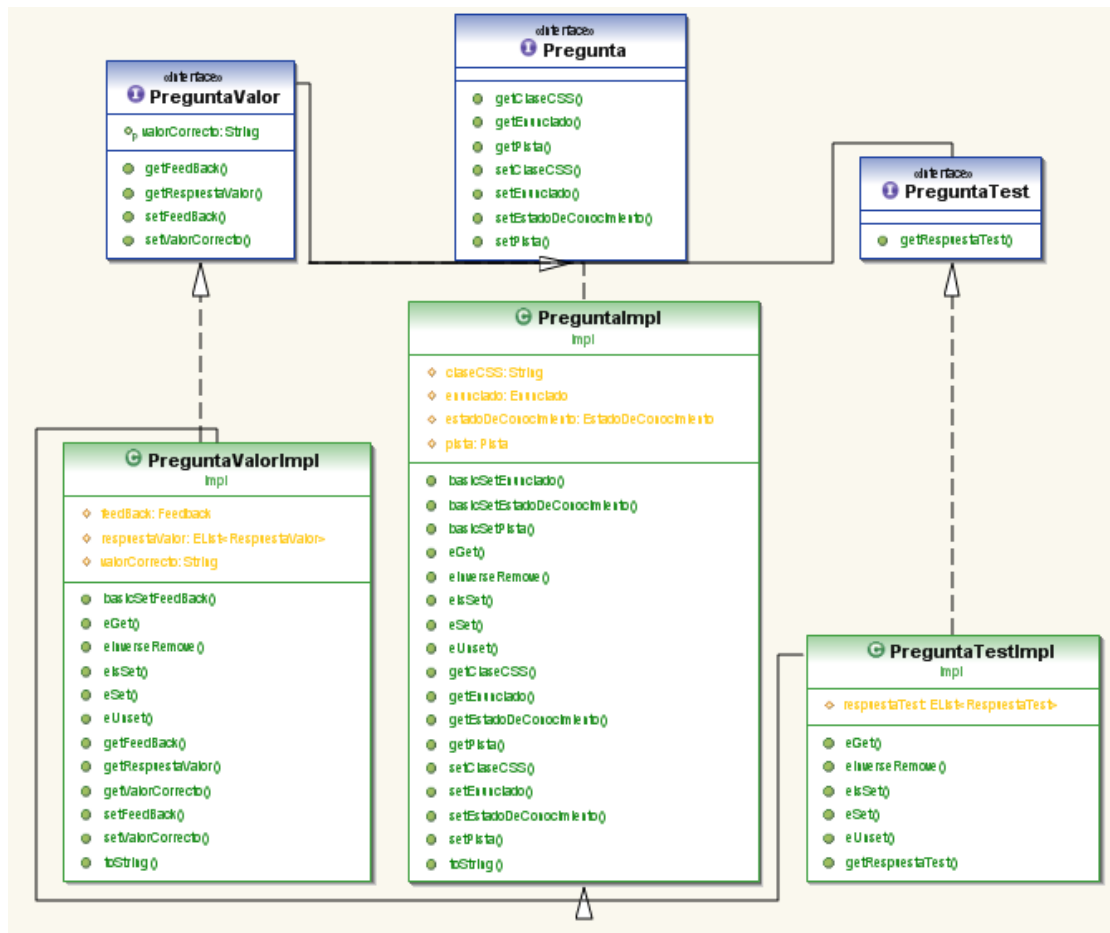


Figura 3.56: Diagrama de clases de *Preguntas*: test y valor

Conocimiento y EstadoDeConocimiento

En la figura 3.59 se muestra la clase *Conocimiento* que encapsula un concepto y nivel. En la figura 3.60 se muestra la clase *EstadoDeConocimiento* se encarga de exponer un conjunto de conocimientos.

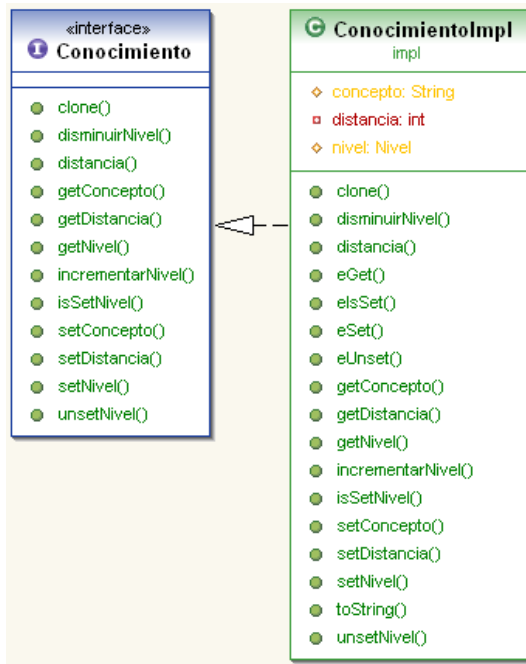


Figura 3.59: Diagrama de clases de *Conocimiento*

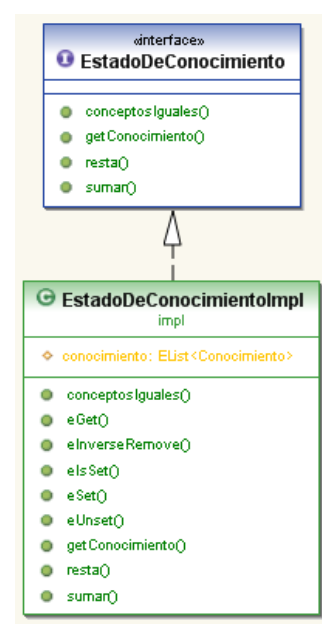


Figura 3.60: Diagrama de clases de *EstadoDeConocimiento*

ConocimientoException

En la figura 3.61 se muestra la clase *ConocimientoException*, que lleva el control de las excepciones del conocimiento.



Figura 3.61: Diagrama de clases de *ConocimientoException*

ErrorNavegacion y CadenaErroresNavegacion

En la figura 3.62 se muestra la clase *ErrorNavegacion*, que gestiona los errores de navegación, devolviendo el mensaje, origen del error y tipo de error que se produce. En la figura 3.63 se muestra la clase *CadenaErroresNavegacion* devuelve la acumulación de errores, cuando se valida la navegación.

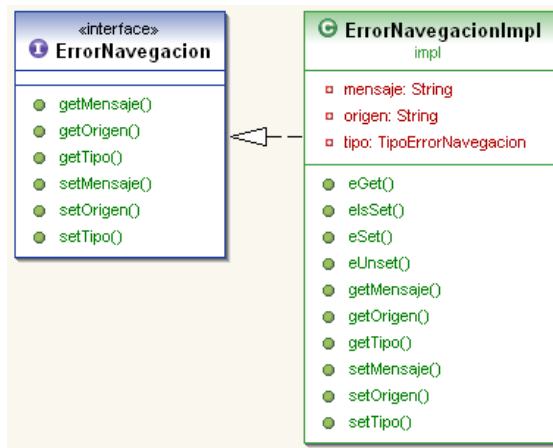


Figura 3.62: Diagrama de clases de *ErrorNavegacion*

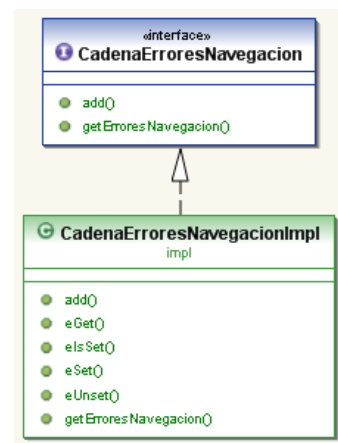


Figura 3.63: Diagrama de clases de *CadenaErroresNavegacion*

3.3.2. Diagrama de clases: Modelo de Datos de las estructuras HTML

El modelo de datos diseñado, para las estructuras HTML, consta de 4 clases: DocumentRoot, Estructuras, Estructura y ParamType.

DocumentRoot

En la figura 3.64 se muestra la clase *DocumentRoot* que se encarga de devolver la raíz del documento XML.

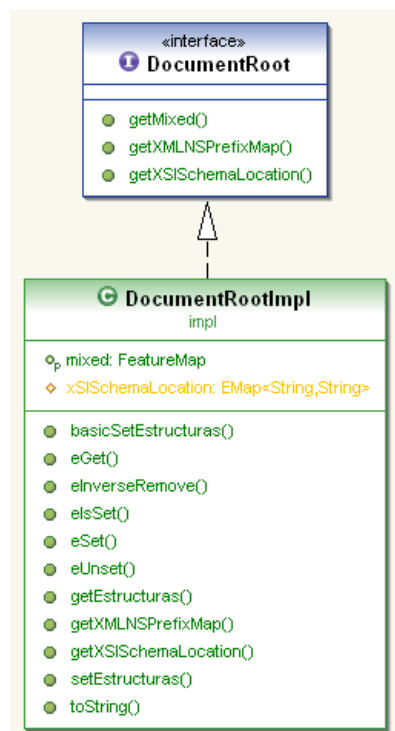


Figura 3.64: Diagrama de clases de *DocumentRoot*

Estructuras

En la figura 3.65 se muestra la clase *Estructuras* que devuelve cada estructura, y la clase *Estructura* que construye el código HTML y devuelve los parámetros que utiliza.

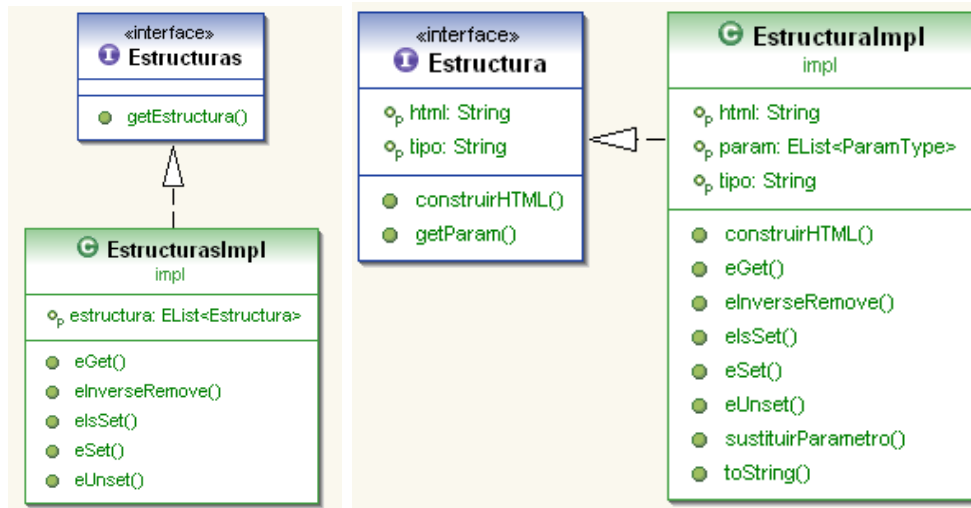


Figura 3.65: Diagrama de clases de *Estructuras* y *Estructura*

ParamType

En la figura 3.66 se muestra la clase *ParamType* que devuelve el nombre y valor que lo conforman. Esta clase se utiliza para especificar cada parámetro de las estructuras HTML.

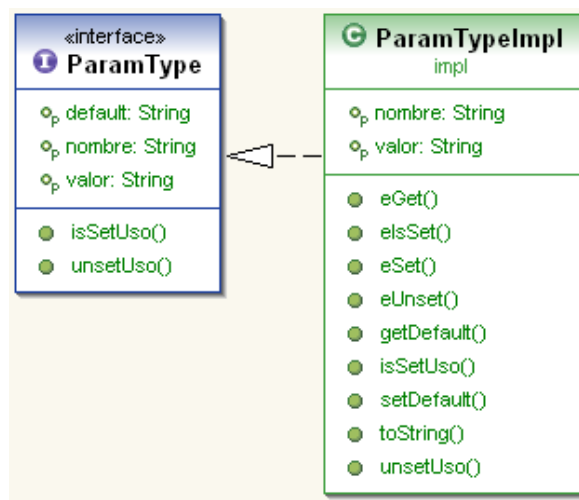


Figura 3.66: Diagrama de clases de *ParamType*

3.3.3. Diagrama de clases: Editor

El diseño del *editor* contiene 28 clases. A continuación, se hace una descripción breve de cada una.

BaseEdicion, RespuestaEdicion y ParrafoEdicion

En la figura 3.67 se muestran la jerarquía de clases utilizadas para la edición en los diálogos de los componentes de un paso.

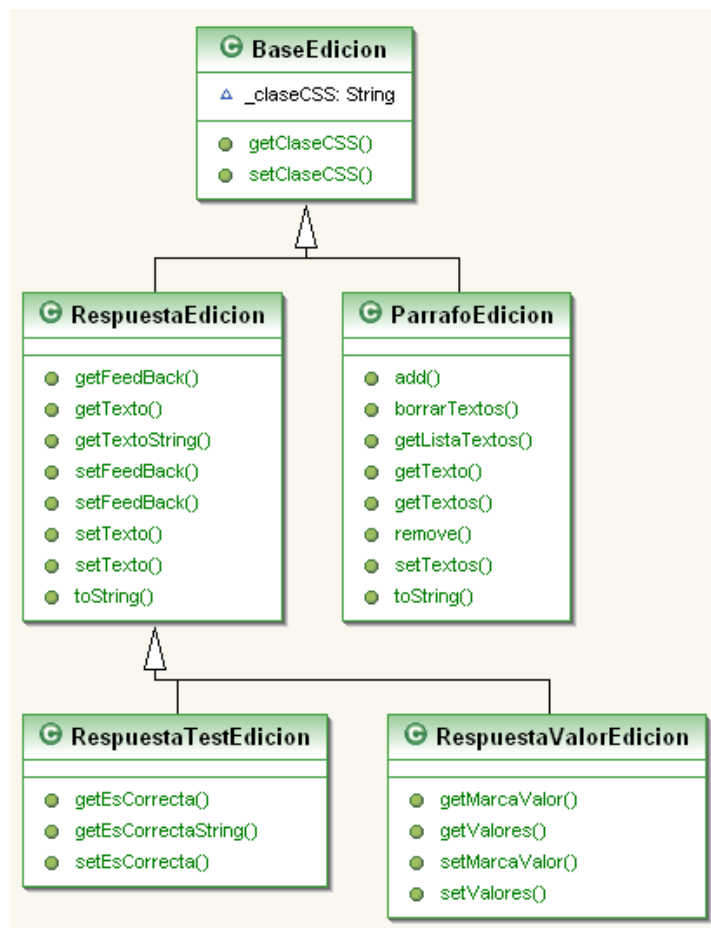


Figura 3.67: Diagrama de clases de *BaseEdicion*, *RespuestaEdicion*(test/valor) y *ParrafoEdicion*

CNarrativoDialog, CMultimediaDialog y CEvaludadorDialog

En la figura 3.68 se muestran las clases para la edición de los diálogos de cada componente: narrativo, multimedia y evaluador.

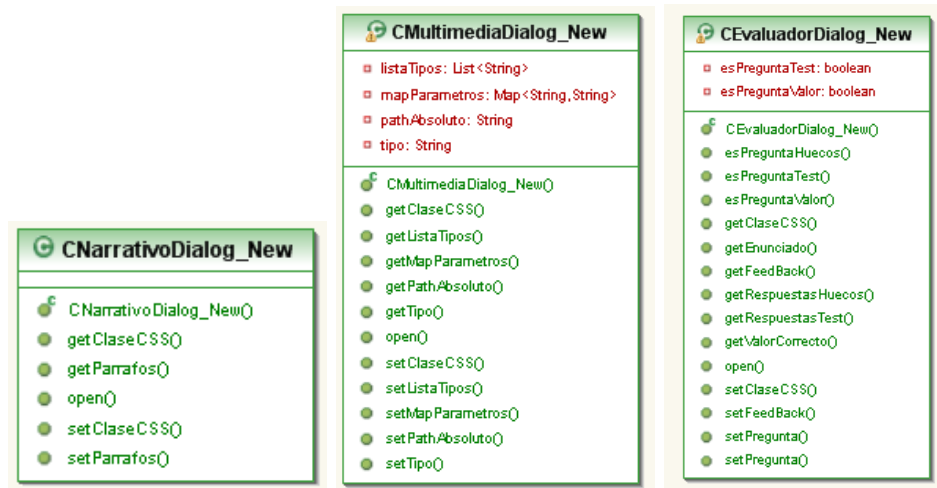


Figura 3.68: Diagrama de clases de *CNarrativoDialog*, *CMultimediaDialog* y *CEvaludadorDialog*

ParametroEdicion y ValorParametroEditingSupport

En la figura 3.69 se muestra la clase *ParametroEdicion* que se encarga de editar los parámetros de las estructuras HTML. En cambio, la clase *ValorParametroEditingSupport*, que se encarga de editar, sobre la interfaz de usuario de edición, el valor de un parámetro.

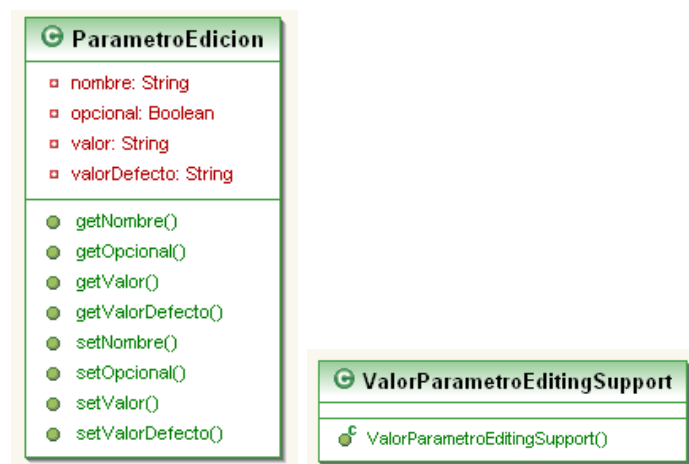


Figura 3.69: Diagrama de clases de *ParametroEdicion* y *ValorParametroEditingSupport*

EditarTextoDialog, EditorTextoConCssDialog y EditorParrafoDialog

En la figura 3.70 se muestra las clases que permiten editar texto, texto con estilo y párrafos con un diálogo de edición.

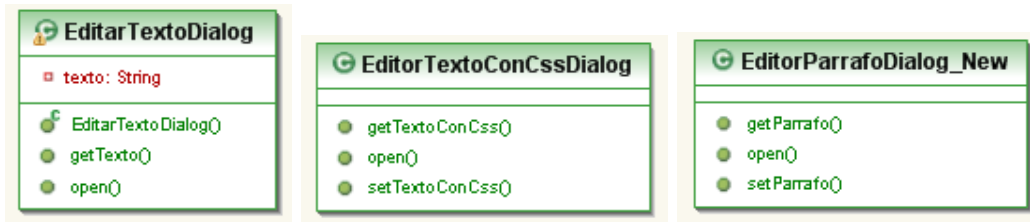


Figura 3.70: Diagrama de clases de *EditarTextoDialog*, *EditorTextoConCssDialog* y *EditorParrafoDialog*

TextoLabelProvider y CssLabelProvider

En la figura 3.71 se muestran las clases que permiten mostrar el valor de un texto o texto con estilo, en la interfaz de usuario.



Figura 3.71: Diagrama de clases de *TextoLabelProvider* y *CssLabelProvider*

EditorRespuestaTestDialog y EditorRespuestaValorDialog

En la figura 3.72 se muestra las clases que permiten editar, mediante un diálogo de edición, las repuestas de tipo test y valor.

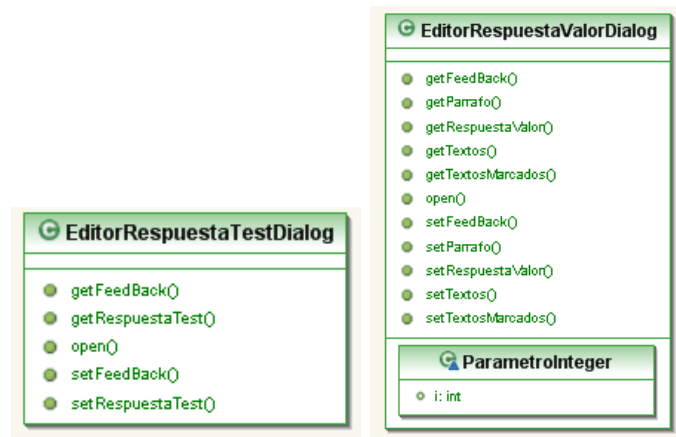


Figura 3.72: Diagrama de clases de *EditorRespuestaTestDialog* y *EditorRespuestaValorDialog*

ConceptoEdicion

En la figura 3.73 se muestra la clase *ConceptoEdicion* que se encarga de editar un concepto.

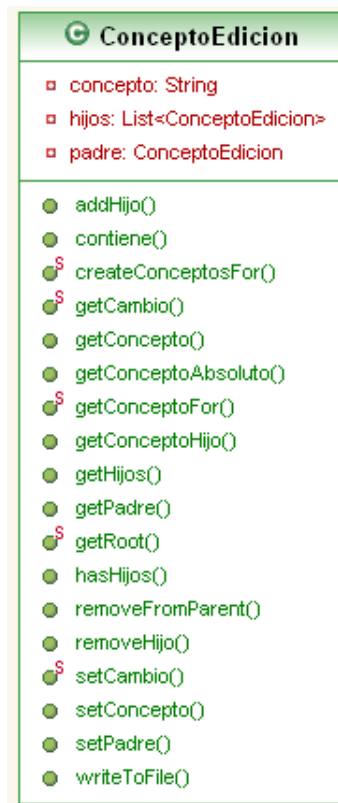


Figura 3.73: Diagrama de clases de *ConceptoEdicion*

ConceptosTreeContentProvider y ConceptoTreeLabelProvider

En la figura 3.74 se muestran las clases que permiten visualizar conceptos en la interfaz de usuario.



Figura 3.74: Diagrama de clases de *ConceptosTreeContentProvider* y *ConceptoTreeLabelProvider*

ConceptoDragListener y ConceptoDropAdapter

En la figura 3.75 se muestra las clases que permiten la funcionalidad de arrastrar y soltar en la interfaz de usuario de edición de los conceptos.

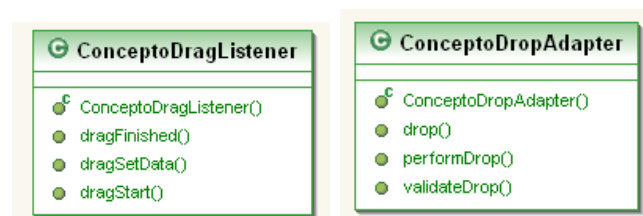


Figura 3.75: Diagrama de clases de *ConceptoDragListener* y *ConceptoDropAdapter*

ConceptoValidator

En la figura 3.76 se muestra la clase *ConceptoValidator* que se encarga de validar un concepto.

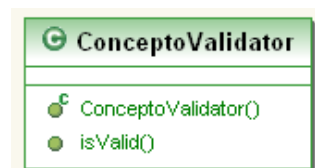


Figura 3.76: Diagrama de clases de *ConceptoValidator*

FeedBackEdicion y EditorFeedBackDialog

En la figura 3.77 se muestra la clase *FeedBackEdicion* que se encarga de editar un feedback, y la clase *EditorFeedBackDialog* que muestra el diálogo de edición de un feedback.



Figura 3.77: Diagrama de clases de *FeedBackEdicion* y *EditorFeedBackDialog*

ConocimientoEdicion y EditorConocimientoDialog

En la figura 3.78 se muestra la clase *ConocimientoEdicion* que se encarga de editar un conocimiento, y la clase *EditorConocimientoDialog* que muestra el diálogo de edición de un conocimiento.

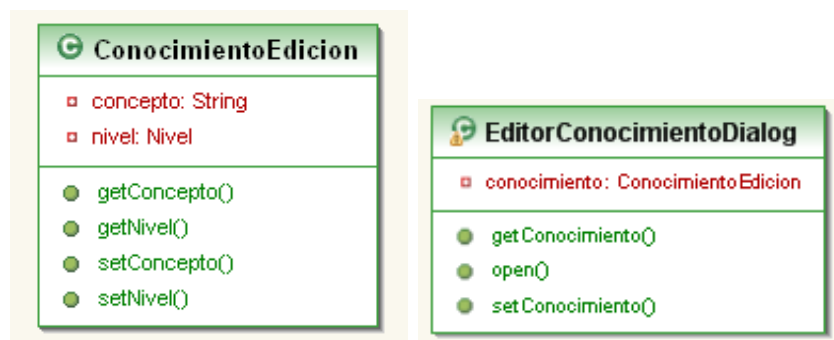


Figura 3.78: Diagrama de clases de *ConocimientoEdicion* y *EditorConocimientoDialog*

EstadoDeConocimientoDialog

En la figura 3.79 se muestra la clase *EstadoDeConocimientoDialog* que permite editar el estado de conocimiento, mediante un diálogo de edición.

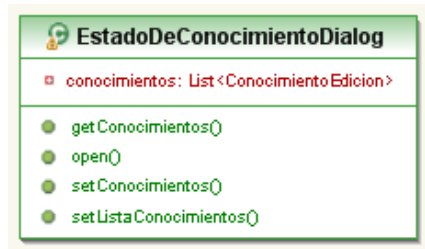


Figura 3.79: Diagrama de clases de *EstadoDeConocimientoDialog*

3.3.4. Diagrama de clases: Repositorio y Librería generador

REPOSITORIO

Para la implementación del patrón de diseño MVC, se han necesitado las clases:

- Formularios, para recoger la información de los documentos .jsp (vista).
- Actions, para realizar las acciones que quiera ejecutar el cliente.
- POJOs, que representan los objetos del modelo de datos.
- DAOs, para hacer consultas/inserciones/modificaciones sobre la base de datos.
- AlmacenPropiedades, para acceder a las propiedades del fichero de configuración.
- Clases de seguridad, para evitar la intrusión en el sistema.
- Clases de validación, para validar a todo aquel que quiera acceder al repositorio Web.

Forms

En la figura 3.80 se muestran las diferentes clases utilizadas para recoger la información de los formularios de las páginas JSP.

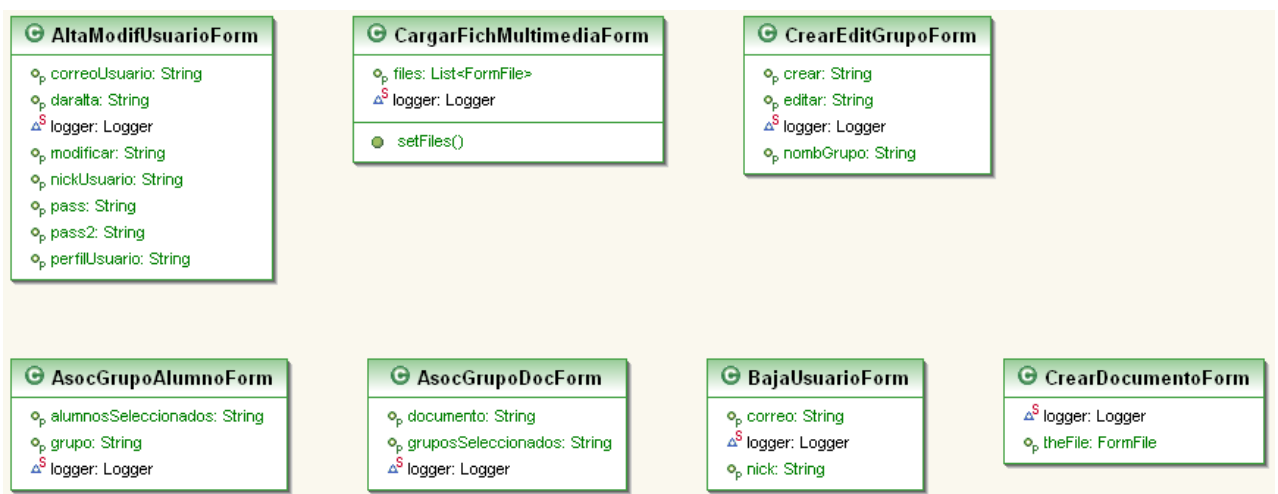


Figura 3.80: Diagrama de clases de *Forms*

Actions

En la figura 3.81 se muestran las diferentes clases utilizadas para, una vez recogida la información de los formularios Jsp, guardar esa información (creando una clase DAO) y devolver la respuesta al usuario.

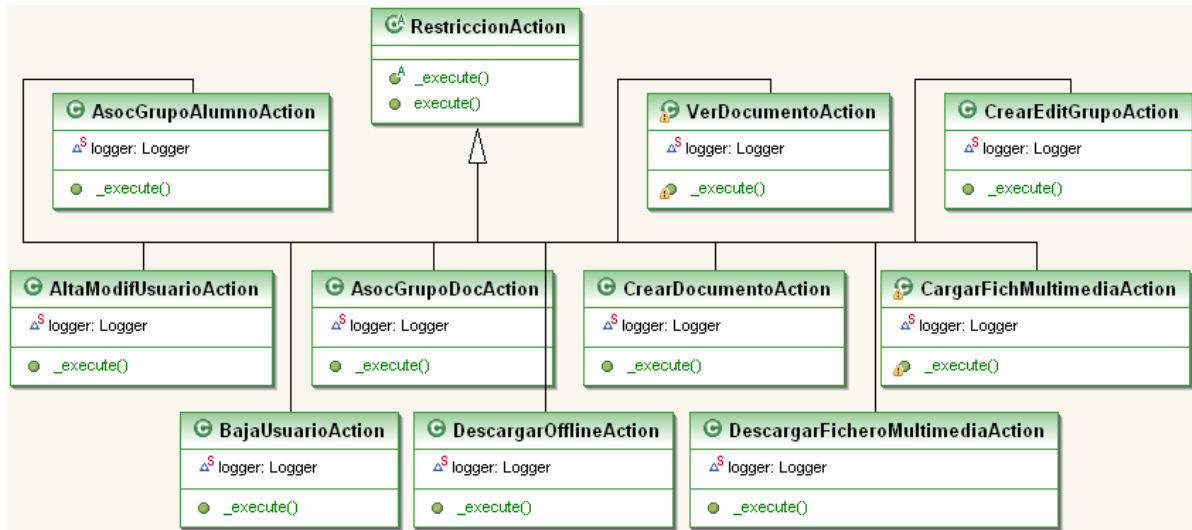
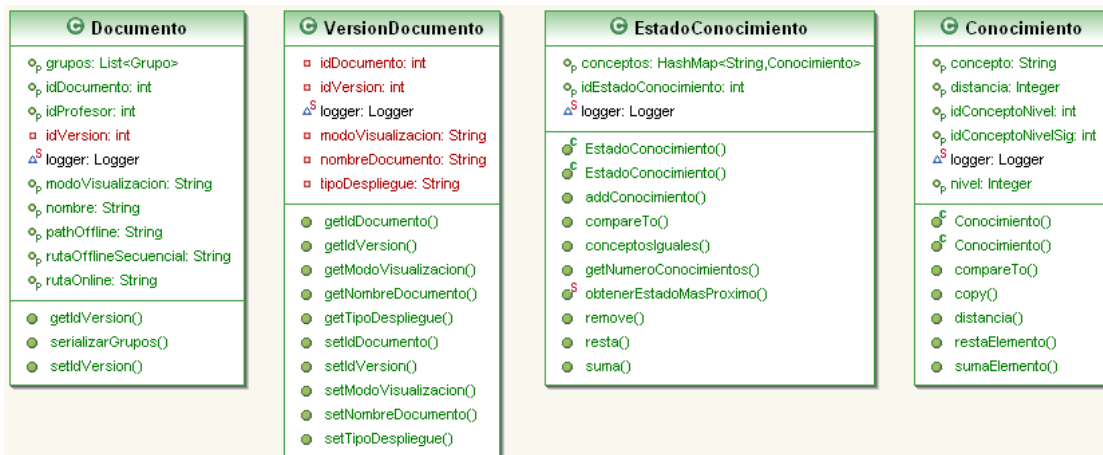


Figura 3.81: Diagrama de clases de *Actions*

POJOs

En la figura 3.82 se muestran las diferentes clases *POJO* que representan los objetos del modelo de datos, y recogen/dan toda la información sobre ellos.



AlmacenPropiedades

En la figura 3.84 se muestran la clase *AlmacenPropiedades* que devuelve las distintas propiedades almacenadas en un fichero de configuración, donde se encuentra el acceso a base de datos y el path de datos del repositorio.



Figura 3.84: Diagrama de clases de *AlmacenPropiedades*

Security

En la figura 3.85 se muestran las diferentes clases de seguridad, creadas para gestionar los accesos al repositorio, evitando intrusiones a las diferentes secciones del repositorio de usuarios sin permisos.



Figura 3.85: Diagrama de clases de *Security*

Validation

En la figura 3.86 se muestran las diferentes clases de validación, creadas para comprobar que el usuario y contraseña sean correctos.

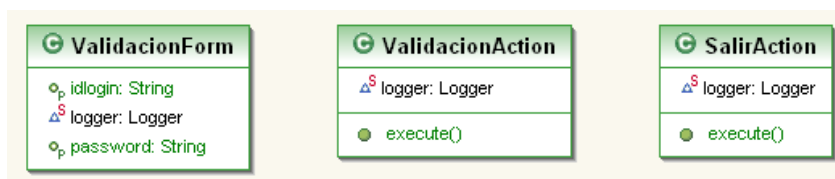


Figura 3.86: Diagrama de clases de *Validation*

LIBRERÍA GENERADOR

La librería generador se compone de 14 clases: *GeneradorDocumento*, *GeneradorPaso*, *GeneradorHTML*, *PaginaWeb*, *PaginaWebOnline*, *AlmacenPropiedades*, *AlmacenEstructuras*, *Utils*, *ResourceDocumento*, *ElementoHTML*, *FaltaPropiedadException* y *Messages*.

Estas clases junto con las de *gestionDocumentos*: *WrapperDocAlumno*, *WrapperGrupo*, *WrapperDoc*, *DController*, *Utils* y *GestionDocumentosException*, dan lugar a la generación y visualización de los documentos offline y online.

Generadores

En la figura 3.87 se muestra la clase de generación de documentos (offline y online), donde se lee el documento XML, creándose las páginas HTML y copiándose todos los archivos necesarios.

Además de la clase de generación de pasos, donde se añaden los componentes: narrativo, multimedia y evaluador.

Y la clase de generación de código HTML, que realmente genera las páginas ayudándose de la clase *ElementoHTML*.

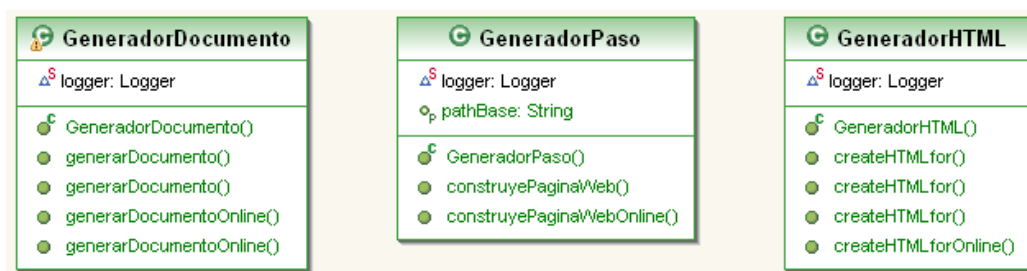


Figura 3.87: Diagrama de clases de *GeneradorDocumento*, *GeneradorPaso* y *GeneradorHTML*

PaginaWeb y PaginaWebOnline

En la figura 3.88 se muestran las clases *PaginaWeb* y *PaginaWebOnline*, modelan la estructura de una página Web, permitiendo su manipulación, tanto para la generación offline como online.



Figura 3.88: Diagrama de clases de *PaginaWeb* y *PaginaWebOnline*

AlmacenPropiedades y AlmacenEstructuras

En la figura 3.89 se muestra la clase *AlmacenPropiedades*, que devuelve las distintas propiedades almacenadas en un fichero de configuración, donde se encuentran los paths necesarios para la generación. La clase *AlmacenEstructuras* que lee el fichero de estructuras de los componentes multimedia, que podrán ser modificados en el futuro, cuando cambien los navegadores Web.

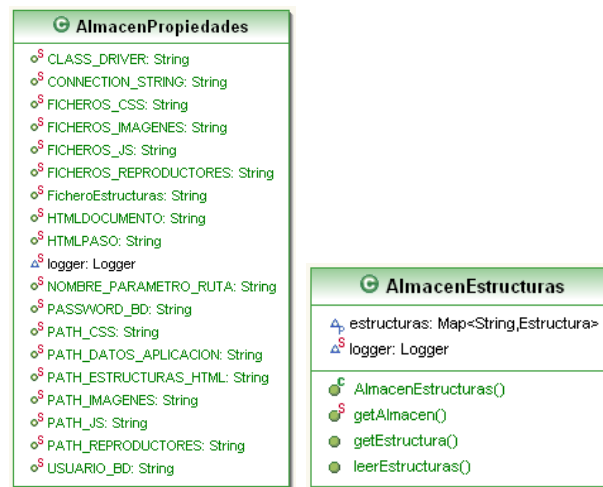


Figura 3.89: Diagrama de clases de *AlmacenPropiedades* y *AlmacenEstructuras*

Utils

En la figura 3.90 se muestran las clases *Utils*, que tiene diferentes funcionalidades para ayudar a la generación de documentos, *ElementoHTML*, clase modelado del código HTML y JavaScript, y *ResourceDocumento*, asocia cada documento con su localización física.

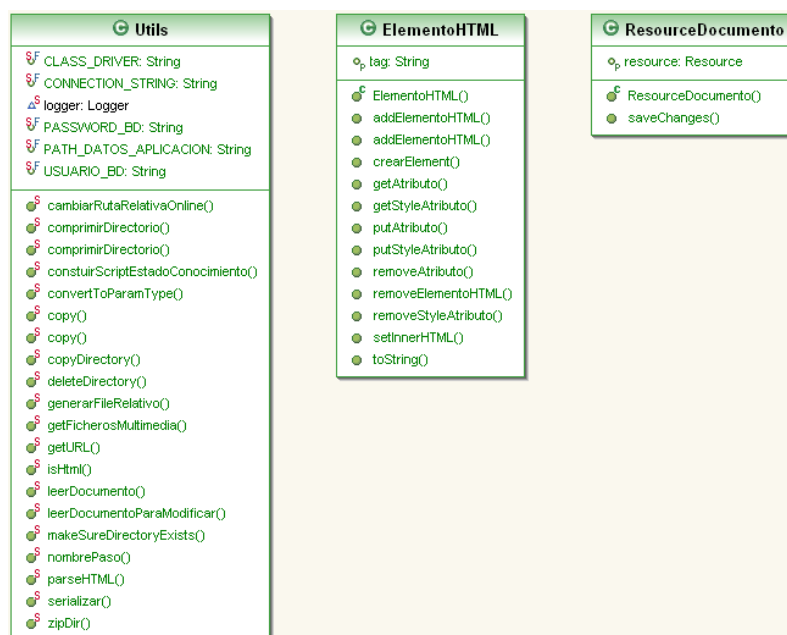


Figura 3.90: Diagrama de clases de *Utils*, *ElementoHTML* y *ResourceDocumento*

Otros

En la figura 3.91 se muestran las clases *FaltaPropiedadException*, que devuelve un mensaje de error, y *Messages*, que devuelve aquellos mensajes almacenados en un fichero de configuración.

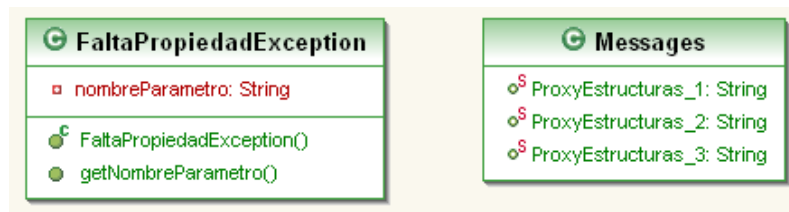


Figura 3.91: Diagrama de clases de *FaltaPropiedadException* y *Messages*

IDocumentoDAO

En la figura 3.92 se muestra la interfaz *IDocumentoDAO*, que tiene los métodos necesarios para serializar un documento.

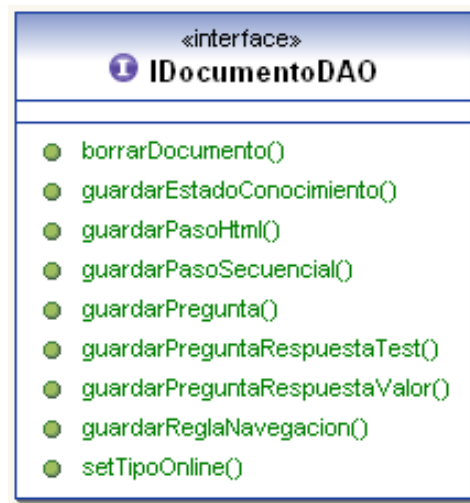


Figura 3.92: Diagrama de clases de *IDocumentoDAO*

Gestión documentos: Wrappers

En la figura 3.93 se muestran las clases *WrapperDoc*, *WrapperGrupo* y *WrapperDocAlumno*, que permiten lanzar las peticiones de las tablas ordenadas, de las páginas JSP, al repositorio.

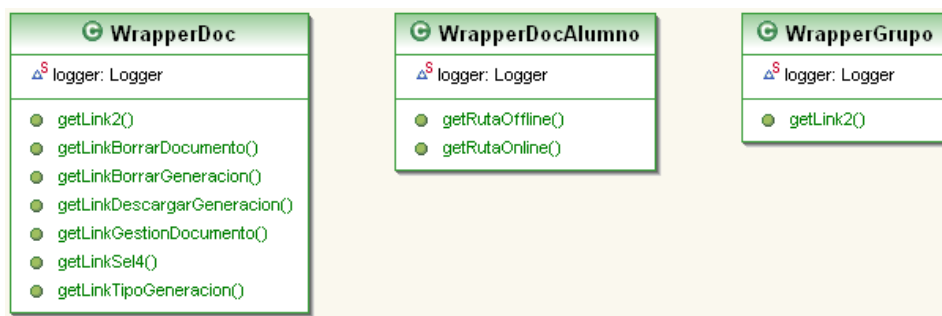


Figura 3.93: Diagrama de clases de *WrapperDoc*, *WrapperGrupo* y *WrapperDocAlumno*

Gestión documentos: DController

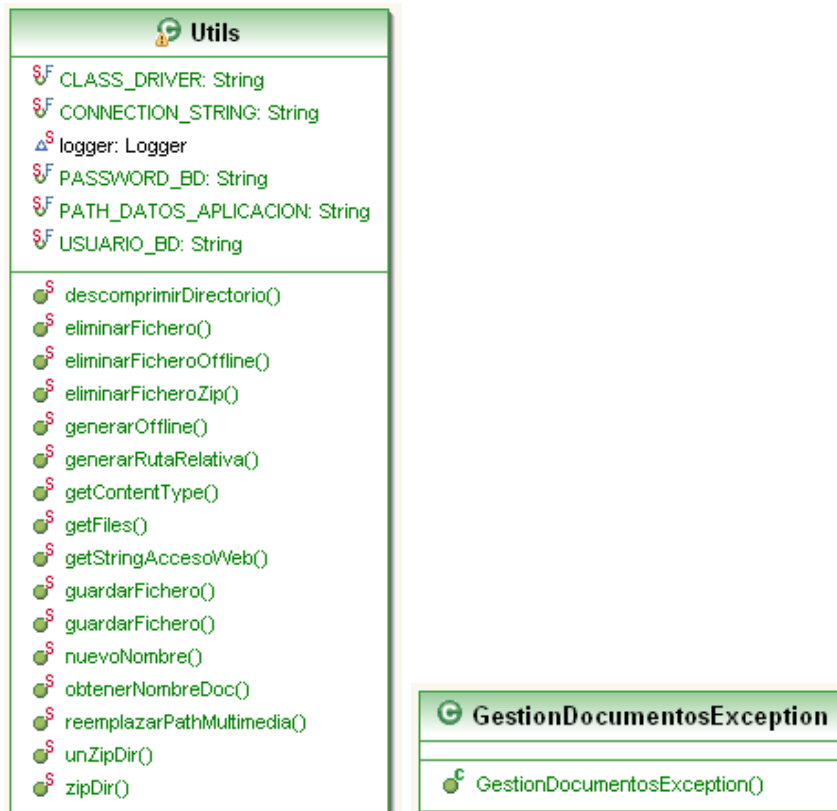
En la figura 3.94 se muestra la clase *DController* que permite la conexión de las páginas JSP (vista) con el servidor (modelo), el cual permite la generación y visualización online.



Figura 3.94: Diagrama de clases de *DController*

Gestión documentos: Utils y GestionDocumentosException

En la figura 3.95 se muestran las clases *Utils*, que contiene distintas funciones que ayudan a la generación offline y online y su visualización, y *GestionDocumentosException*, que capturas excepciones.

Figura 3.95: Diagrama de clases de *Utils* y *GestionDocumentosException*

3.3.5. Diseño de la Base de Datos

Tanto el repositorio como la generación y visualización de documentos formativos han necesitado una base de datos, para guardar toda la información. Para el diseño y almacenamiento de la base de datos se ha usado la herramienta SQLyog [29], utilizando la base de datos MySQL [33], por su facilidad de uso y para permitir un futuro mantenimiento, además de estar desarrollada como software libre.

Para su implementación se ha diseñado un modelo ER, que se muestra a continuación.

3.3.5.1. Diagramas Entidad-Relación

En este apartado se van a mostrar los diagramas ER, que muestran el diseño de la base de datos. Estos diagramas muestran las relaciones existentes entre las diferentes tablas de la base de datos, tanto para el repositorio como para la generación y visualización de los documentos offline y online.

Repositorio

Admin

La figura 3.96 muestra que el usuario, que pertenece a la sección *admin*, puede crear, modificar y eliminar usuarios.

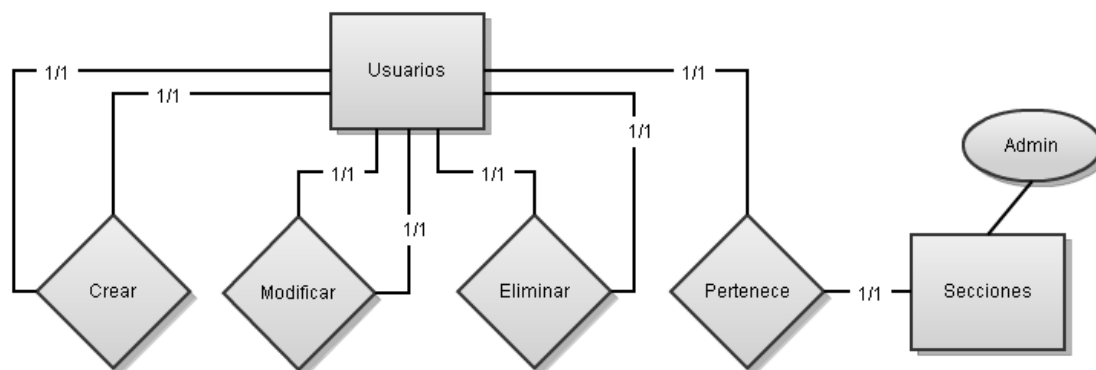
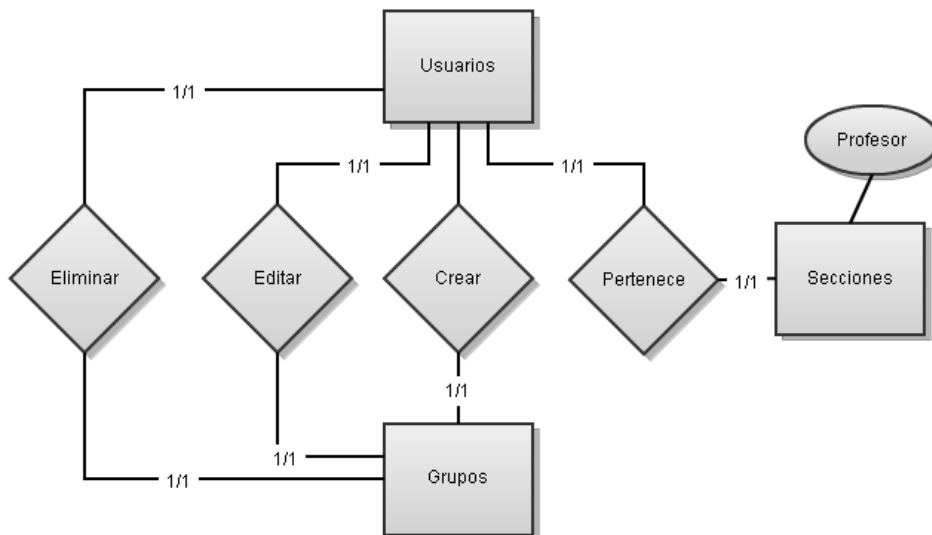


Figura 3.96: Diagrama de E/R del **repositorio**: usuario *Admin*

Profesor: Gestión grupos

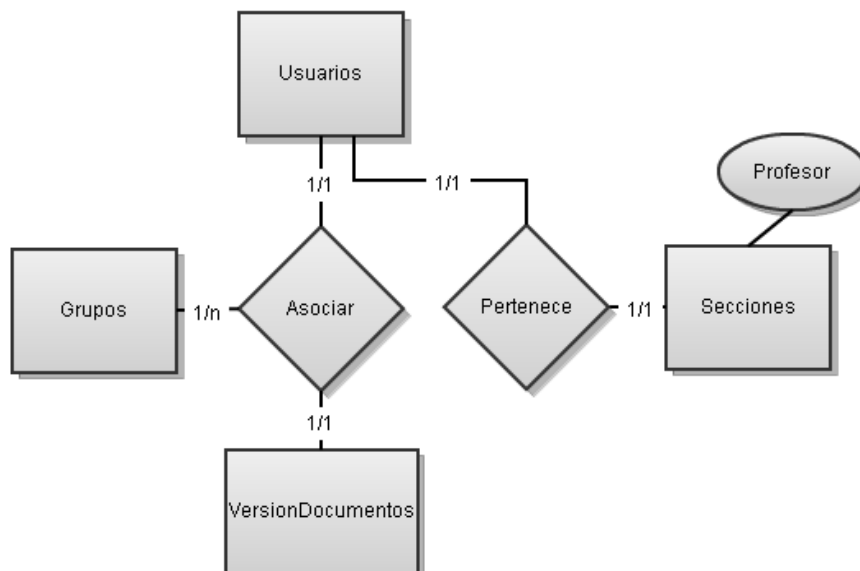
Además, este usuario puede crear, editar y eliminar grupos.



(b) Docente: Gestión grupos

Profesor: Asociación documento y grupos

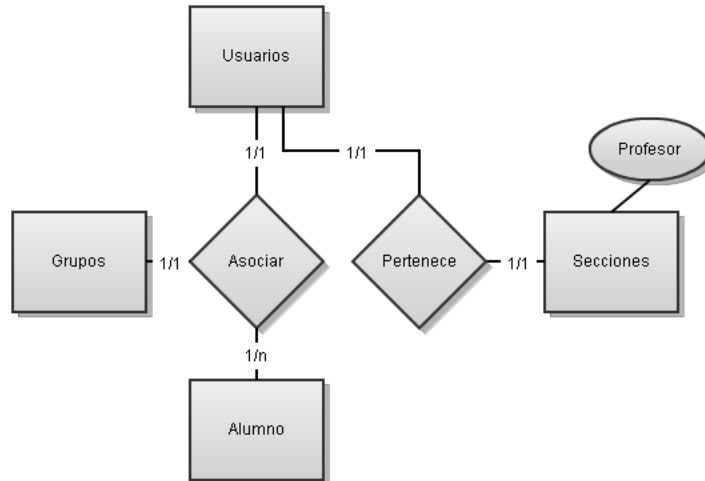
Este usuario puede asociar cada versión de un documento a uno o varios grupos.



(c) Docente: Asociación documento y grupos

Profesor: Asociación grupo y alumnos

Por último, este usuario puede asociar alumnos a un grupo.



(d) Docente: Asociación grupo y alumnos

Figura 3.98: Diagrama de E/R del repositorio: usuario *Profesor*

Generación y Visualización de documentos

La figura 3.99 muestra que el usuario, que pertenece a la sección *profesor*, puede generar una versión de documento offline, que pertenece a un documento, y ambos están asociados a uno o varios grupos.

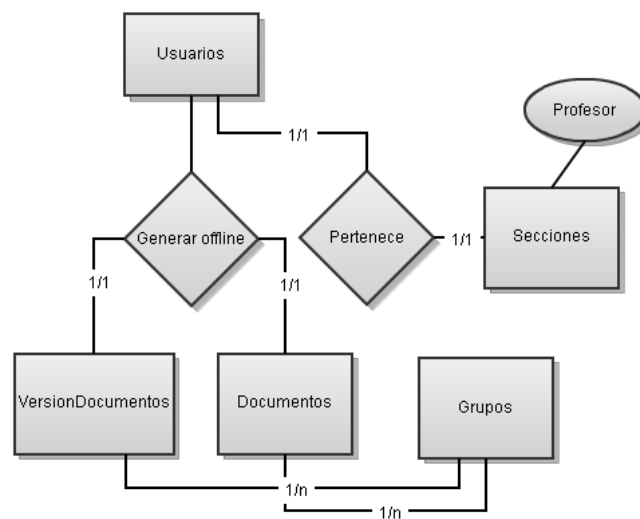


Figura 3.99: Diagrama de E/R de la *generación de documentos offline*

Además, la figura 3.100 muestra que este usuario puede generar una versión de documento online. Esta generación insertará: el estado de conocimiento de cada paso y de las reglas de navegación, el código html de los pasos, la lista de pasos secuencial, y las preguntas y respuestas del documento.

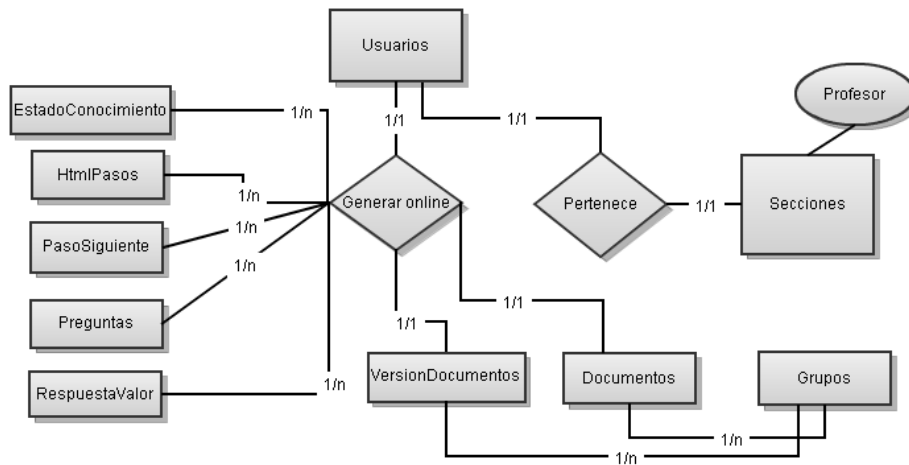


Figura 3.100: Diagrama de E/R de la *generación de documentos online*

Por último, la figura 3.101 muestra que este usuario puede visualizar versiones de documentos online. Esta visualización necesitará sacar de la base de datos: el estado de conocimiento de pasos y reglas, el código html de los pasos, la lista de pasos secuencial, y las preguntas y respuestas del documento.

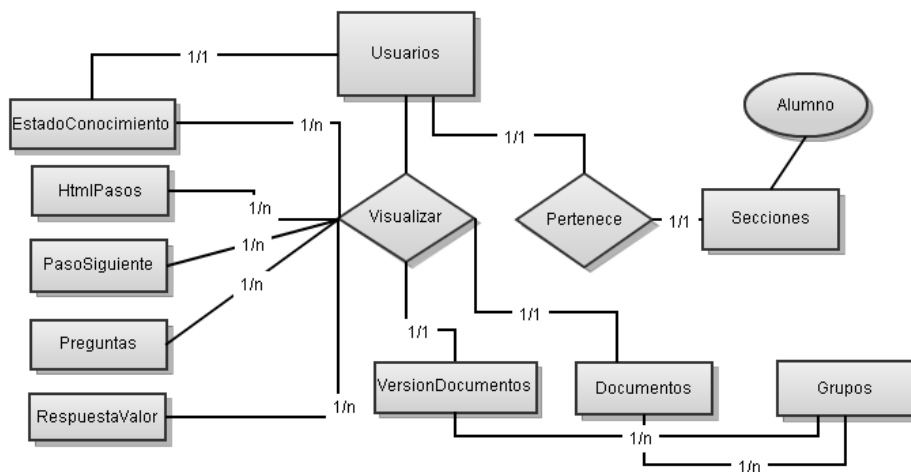


Figura 3.101: Diagrama de E/R de la *visualización de documentos online*

3.3.5.2. Esquemas Entidad-Relación

En este apartado se van a mostrar los distintos esquemas ER, donde se explicará que contiene cada tabla dentro de la base de datos.

Repositorio

- **Grupos y documentos:**

La tabla *documentos* se encarga de almacenar los documentos añadidos al repositorio y, una vez realizada su generación offline, almacena la ruta donde se encuentra dentro del servidor Web.

La tabla *versiondocumentos* se encarga de almacenar las versiones de los documentos insertados, su tipo de despliegue (offline/online) y su modo de visualización (secuencial/adaptativa).

La tabla *grupos* almacena los diferentes grupos o asignaturas que se han creado en la repositorio, así como el *identificador* del profesor que tiene documento/s asociado/s a ese grupo.

La tabla *grupodoc* almacena la asociación de grupos, documentos y versiones de documentos.

La tabla *tipogeneracion* almacena los tipos de generación que existen.

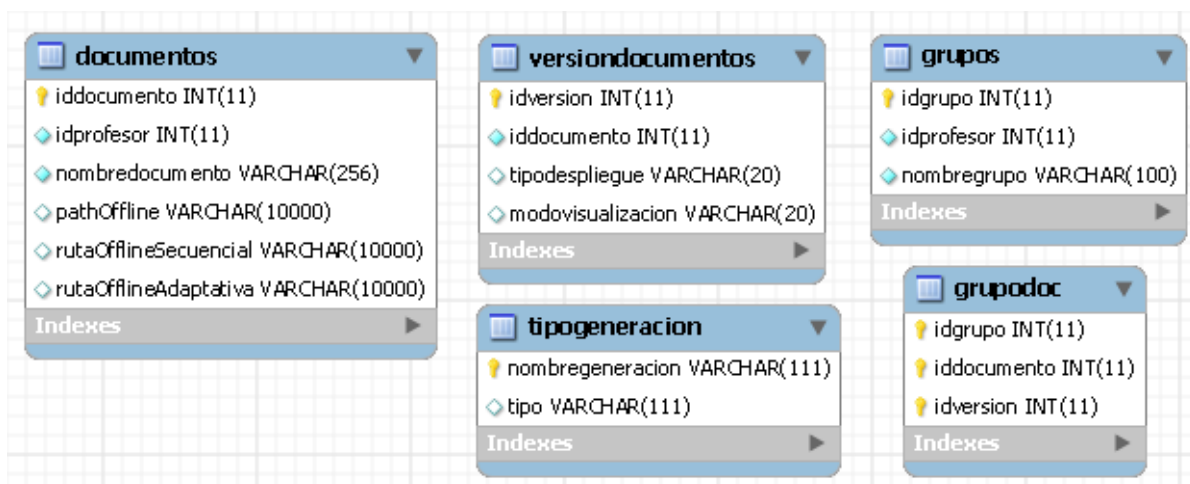


Figura 3.102: Diagrama de E/R de documentos, versiondocumentos y grupos

- **Grupos y alumnos:**

La tabla *usuarios* se encarga de almacenar la información de los usuarios del repositorio. Además, de cada usuario, contiene el *identificador* de su estado de conocimiento.

La tabla *alumnogrupo* almacena la asociación de alumnos y grupos.

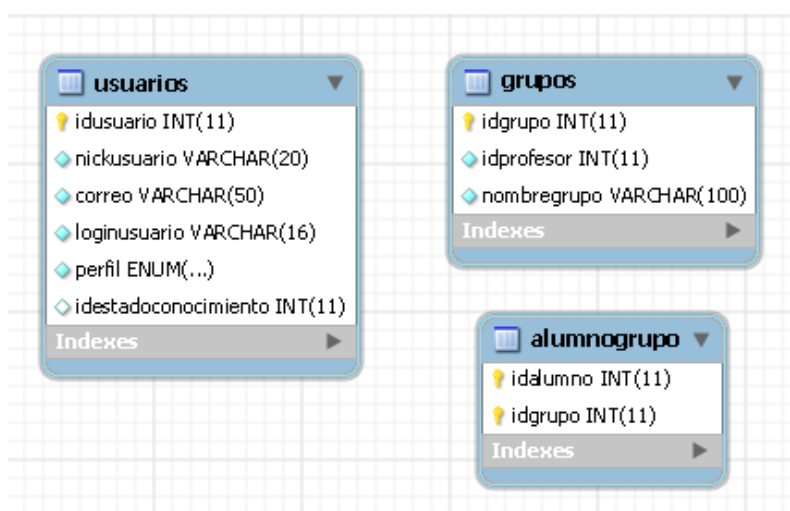


Figura 3.103: Diagrama de E/R de grupos y alumnos

Generación y visualización de documentos

- **Navegación secuencial:**

La tabla *preguntas* se encarga de almacenar todas las preguntas de los documentos añadidos en el repositorio. Conteniendo el *identificador* del documento, del paso asociado y del estado de conocimiento de la pregunta.

La tabla *respuestavalor* almacena todas las respuestas de las preguntas. Conteniendo el *identificador* de la pregunta asociada.

La tabla *pasosiguiente* almacena los pasos secuenciales de los documentos añadidos en el repositorio.

La tabla *htmlpasos* almacena el código HTML, para la visualización online.

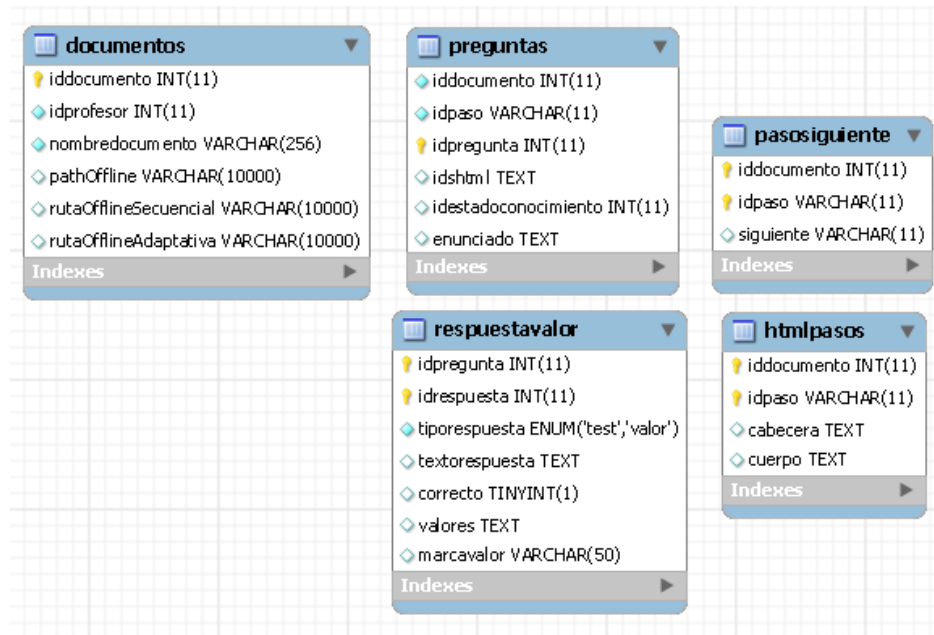


Figura 3.104: Diagrama de E/R de documentos offline y online secuenciales

■ Navegación adaptativa:

La tabla *reglaspasos* se encarga de almacenar las reglas de navegación. Además, contiene un *identificador* del estado de conocimiento que se puede alcanzar a través de ellas.

La tabla *estadoconocimiento* almacena el estado de conocimiento de los alumnos y de las reglas de navegación.

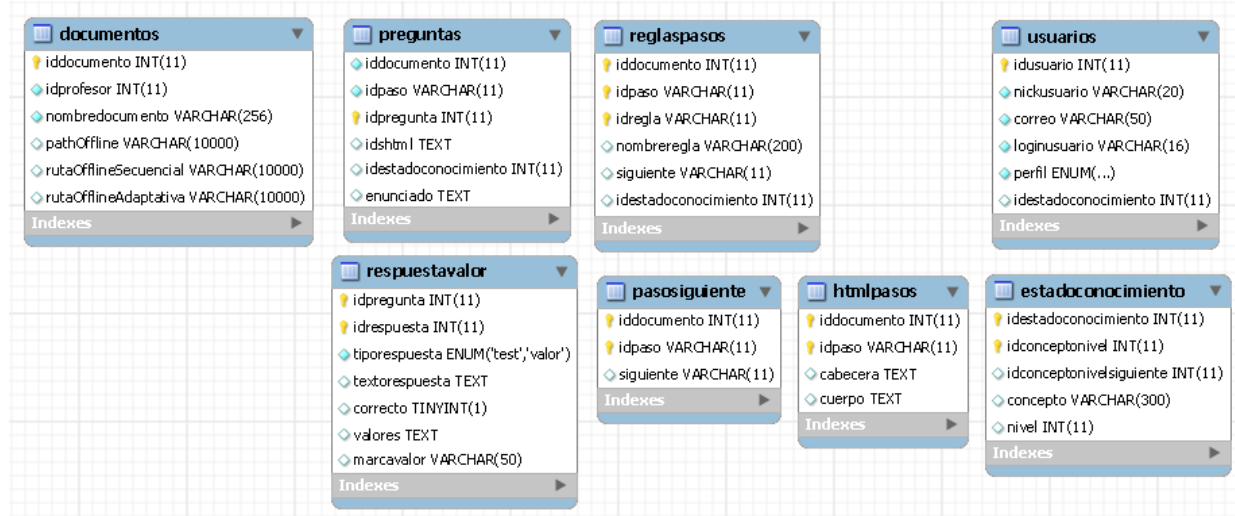


Figura 3.105: Diagrama de E/R de documentos offline y online adaptativos

3.4. IMPLEMENTACIÓN

La implementación de este proyecto se ha dividido en dos partes, una referente al *editor* y otra al repositorio y *generación y visualización de documentos*.

Durante el proceso de implementación del *editor de documentos* se han respetado los paradigmas de: la programación orientada a objetos (abstracción, encapsulación, modularidad y jerarquía); de la programación orientada a eventos (cada componente reacciona ante los eventos producidos cuando el usuario interactúa con la aplicación, con métodos asociados para cada evento) y el de programación estructurada dentro de los métodos de las clases. Se han seguido las recomendaciones de codificación y las normas de estilo de Java, como que los nombres de clases comienzan en mayúsculas y los nombres de variables y métodos en minúsculas.

En cuanto a la implementación del repositorio, y *generación y visualización de documentos* también se han respetado los paradigmas de la programación orientada a objetos, y el de programación estructurada dentro de los métodos de las clases. Además de seguir las normas de estilo de Java.

Editor

En la implementación del *editor de documentos* se ha utilizado el entorno de desarrollo Eclipse, junto con las bibliotecas gráficas SWT [13] y JFace [12]. Gracias a la utilización de JFace, el diseño de esta aplicación de escritorio se basa en el patrón MVC, permitiendo un desarrollo más estructurado y con mayor robustez.

Repositorio

En la implementación del repositorio se ha utilizado la combinación de los frameworks Struts [14] y Tiles [16], junto con los lenguajes de programación JSP [20] (vista) y Java [34] (modelo), estos han obligado a configurar *struts-config.xml*, *web.xml* y *tiles-*

defs.xml. Struts utiliza su fichero de configuración para separar el modelo, la vista y el controlador, y que la implementación quede estructurada. Y, Tiles, para ayudarnos a que la implementación de Struts sea menos engorrosa.

Generación y visualización de documentos formativos

Generación offline

La generación de documentos offline se ha llevado a cabo a través de la librería Java implementada para ello. En esta librería se modela la estructura HTML, de cada uno de los pasos, mediante clases propias que facilitaban la construcción del árbol DOM [28], *ElementoHtml*. Este (ElementoHTML) permite el modelado o inclusión del código JavaScript [25], estilos, componentes multimedia, narrativos y evaluador. Una vez generados todos los pasos, y copiados los elementos multimedia, reproductores, estilos y otros, la librería comprime el directorio creado, con una clase de Java *ZipOutputStream*, consiguiendo así el archivo que luego el alumno descargará.

Generación online

La generación de documentos online, desde el repositorio, comienza con la llamada desde JavaScript al controlador DWR [22], que llamará a la misma librería Java que en offline, pero con ciertos matices. Esta librería, para la generación online, modela también la estructura HTML, de cada uno de los pasos, mediante la clase *ElementoHtml*. Pero no incluye JavaScript destinado a la navegación offline, ni tampoco toma los estilos, simplemente añade los componentes multimedia, narrativos y evaluador que necesita. Una vez generado un paso, es añadido a la base de datos. Además del código Html, se guarda en la base de datos toda la lógica de navegación, y las preguntas con sus correspondientes respuestas.

Visualización offline

La visualización de documentos offline se lleva a cabo totalmente a través de JavaScript, que lleva el control de las preguntas, respuestas y estado de conocimiento del

alumno, y la navegación (secuencial/adaptativa).

Visualización online

La visualización de documentos online utiliza también JavaScript, que llama al controlador DWR. Este creará los pasos y llevará el control en sesión tanto del estado de conocimiento del alumno y la navegación, como del resto de información necesaria para el correcto funcionamiento. Además, la creación de los pasos requiere de conexión con la base de datos, para obtener toda la información guardada en el proceso de generación online.

3.4.1. Diferencias: requisitos y resultado

3.4.1.1. Requisitos iniciales del sistema

Editor

El objetivo principal era la creación y edición de documentos formativos, para la enseñanza de temas concretos a diferentes niveles de profundidad. Generando un archivo estructurado que serviría para su visualización en la Web.

El editor poseería varias vistas, sobre el mismo documento: una para la edición de los pasos, y otra para la navegación entre ellos. En la vista de edición de los pasos, la aplicación permitiría añadir componentes multimedia, narrativos y un sólo evaluador, por paso. Además, de poder editar cada componente:

- Componente narrativo: se añadirían los párrafos.
- Componente multimedia: dependiendo del tipo (imagen, sonido y vídeo), se editarían la url, tamaño, tipo, etc.
- Componente evaluador: se añadirían las preguntas (tipo test, simple o multiopción, un valor correcto o rellenar huecos), y sus respuestas.

Generación y visualización de documentos

El objetivo principal era construir documentos offline y online a partir de archivos estructurados. Estos documentos tendrían una estructura concreta y definida para albergar la definición de todos los pasos que los constituyeran. Cada paso se compondría de una parte gráfica, otra narrativa y opcionalmente otra interactiva.

La parte gráfica sería una representación visual del tema que tratara el documento. Podría ser: imagen, sonido o animación (vídeo o cualquier otro). La narrativa se representaría en texto. La interactiva se compondría de preguntas, donde la pregunta constaría de un enunciado, un conjunto de respuestas (tipo test o respuesta libre) y una ayuda. De esta manera, si el alumno respondiera de manera incorrecta, se mostraría el motivo del fallo (feedback).

Además, dependiendo de la elección del docente, el orden de reproducción de los pasos podría ser secuencial o adaptativo. El secuencial no variaría, en cambio el adaptativo se definiría según los niveles de conocimiento del tema. Este dependería, además, del nivel que el alumno demostrara tener, pudiendo cambiar de nivel, subir o bajar. Así, un camino identificaría el siguiente paso a mostrar dependiendo de la respuesta seleccionada, de su corrección y del nivel actual del alumno.

Por otro lado, dependiendo del tipo de documento: offline y online, este recogería la información en sesión o en base de datos, y sería visualizable sin conexión a Internet o no podría ser visto sin ella. Además, estos documentos tendrían que estar disponibles para el alumno desde una servidor Web, donde este podría descargarse los de tipo offline, y visualizar los tipo online.

3.4.1.2. Resultado final

Editor

El resultado ha sido una herramienta para la creación y edición de estos documentos formativos, que cumple todos los requisitos funcionales y parte de los requisitos no funcionales. Esta aplicación puede mejorar su interfaz gráfica, haciéndola todavía más fácil de usar, además de hacerla más accesible a la mayor parte de las personas [1].

Generación y visualización de documentos

El resultado ha sido la generación de estos documentos formativos a partir de archivos Xml [27]. Previamente, se ha creado un repositorio para gestionar documentos, grupos (asignaturas), alumnos y su asociación. Así, el docente puede gestionar la subida de documentos al servidor, generar los documentos que prefiera, y asociarlos a sus asignaturas, de una forma cómoda y sencilla. Los pasos a seguir para conseguir que estos documentos offline y online estén disponibles por los alumnos son:

- Seleccionar el documento comprimido .zip, previamente creado por el profesor, de su PC, y “Subir documento”.
- Seleccionar de la tabla de *Gestión Documentos* el documento recientemente añadido, y generar en offline: “secuencial” y “adaptativa”, en online: “secuencial” y “adaptativa”.
- Una vez generados los documentos, el docente debe *Asociar grupos y documentos*, donde “Asociar” el nuevo documento al grupo que desee.

Una vez hecho esto, los alumnos podrán acceder a los documentos que el profesor haya generado.

Por otro lado, en cuanto a los componentes, estos han seguido las pautas indicadas en los requisitos iniciales. Pudiendo añadir el número que queramos de componentes narrativos y multimedia en cada paso, pero sólo uno o ningún componente evaluador.

En cuanto al tipo de reproducción adaptativa, primeramente se pensó que el nivel del alumno pudiera subir o bajar, dependiendo de si respondía bien o mal a las preguntas formuladas. Al final, esta idea se rechazó ya que un alumno no puede perder conocimiento de un tema, simplemente deja de adquirirlo, pudiendo sólo aumentar su nivel o mantenerlo. Además, según las reglas de navegación, dependiendo del nivel adquirido del alumno, se navegará a aquel paso con estado de conocimiento más próximo, menor o igual. Esto es, si no hay regla de navegación en ese paso se utilizará la navegación secuencial, pero en el caso que la haya, se tomará aquella con más conceptos, y más próximos al alumno. De esta manera, el alumno irá adquiriendo niveles de conocimiento sobre diferentes conceptos de la asignatura que, al final del documento, podrá visualizar, y si es online se actualizarán en la base de datos.

3.4.2. Plataforma

3.4.2.1. Editor

Para la implementación de este *editor de documentos* se ha utilizado el entorno de desarrollo Eclipse, facilitando en gran medida el desarrollo. Además de otras tecnologías: Java, como lenguaje de programación, SWT y JFace, como bibliotecas gráficas.

Además, para hacer que el editor fuera más fácil de usar y atractivo al usuario se han utilizado los frameworks de Eclipse EMF [8], GEF [10] y GMF [11].

Por otro lado, el editor genera como resultado, entre otros archivos, un documento XML, que seguirá un esquema específico para su validación.

3.4.2.2. Repositorio

Para el diseño del repositorio, se ha utilizado el patrón MC, y para su implementación se han utilizado varias tecnologías:

- Para la interfaz Web se ha utilizado la tecnología JavaServer Pages con Apache Struts y Tiles.
- El servidor utilizado ha sido Apache Tomcat [15].
- La implementación del Modelo de datos se ha hecho con el lenguaje Java.
- Para la conexión a la Base de datos desde Java se ha utilizado JDBC [19].
- Y el lenguaje utilizado para la base de datos MySql [33] es SQL [26].

3.4.2.3. Generación y visualización de documentos

Para la generación y visualización de los documentos offline y online también se han utilizado varias tecnologías.

- Generación de documentos offline:
 - Desde el repositorio, al pulsar en el enlace de generación, se lanzará desde JavaScript una llamada a la librería Java.
 - Desde la librería Java, se generarán los pasos Html y se copiarán los archivos multimedia.
 - Y, por último, se creará el archivo comprimido .zip.
- Generación de documentos online:
 - Desde el repositorio, al pulsar en el enlace de generación, se lanzará desde JavaScript una llamada al controlador DWR.
 - El controlador llamará a la librería Java, que generará los pasos Html, que se guardarán en la base de datos, y se copiarán los archivos multimedia al servidor.

- Visualización de documentos offline:
 - El documento .zip descargado es descomprimido por el alumno y, una vez hecho esto, el alumno podrá ejecutar el documento *index.html*. Toda la interacción con el alumno será a través de JavaScript, y la creación del contenido de los diferentes pasos será gracias a iFrame [24].

- Visualización de documentos online:
 - Desde el repositorio, sección Alumno, al pulsar el documento seleccionado el alumno podrá visualizar el documento.
 - La visualización e interacción de los pasos Html se hará desde JavaScript, que llamará al controlador DWR y este llamará a la librería Java.

Tecnologías utilizadas:

Java [34]

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90.

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución.

Java fue construido desde el principio para ser completamente orientado a objetos. Todo en Java es un objeto (salvo algunas excepciones), y todo en Java reside en alguna clase (una clase es un molde a partir del cual pueden crearse varios objetos).

SWT [13]

SWT [Standard Widget Toolkit] es un conjunto de componentes para construir interfaces gráficas en Java, (widgets) desarrollados por el proyecto Eclipse.

Recupera la idea original de la biblioteca AWT de utilizar componentes nativos, con lo que adopta un estilo más consistente en todas las plataformas, pero evita caer en las limitaciones de ésta.

La interfaz del workbench de eclipse también depende de una capa intermedia de interfaz gráfica de usuario (GUI) llamada JFace.

JFace [12]

JFace es un conjunto de widgets para realizar interfaces de usuario construido sobre SWT. Fue desarrollado por IBM para facilitar la construcción del entorno de desarrollo Eclipse, pero su uso no está limitado a este.

JFace proporciona una serie de construcciones muy frecuentes a la hora de desarrollar GUIs, tales como cuadros de diálogo, evitando al programador la tediosa tarea de lidiar manualmente con los widgets de SWT.

EMF [8]

EMF [Eclipse Modeling Framework], es un framework de modelado, para facilitar la generación de código, para la construcción herramientas y otras aplicaciones basadas en un modelo de datos estructurado.

Desde una especificación del modelo descrita en XMI, EMF da soporte para producir clases Java para el modelo, conjunto de clases *Adapter* que permiten visualización y edición, y un editor básico.

GEF [10]

GEF [Graphical Editing Framework], es un framework que fue desarrollado para la plataforma Eclipse. Se utiliza para crear editores gráficos de diferentes esquemas como: redes eléctricas o diagramas en árbol. Estos diagramas ofrecen capacidades de edición fácil para los datos en dominios específicos y los adaptan a una representación gráfica.

Generalmente, GEF es usado como parte del framework de modelado gráfico GMF, que combina EMF y GEF para crear el código necesario para el modelo de datos y el editor de diagramas.

GMF [11]

GMF [Graphical Modeling Framework], es un framework dentro de la plataforma Eclipse. Proporciona un componente de generación e infraestructura, en tiempo de ejecución, para el desarrollo de editores gráficos basado en los frameworks de modelado de Eclipse (EMF) y editor gráfico (GEF).

XML [27]

XML [eXtensible Markup Language] desarrollado por el World Wide Web Consortium (W3C), es un lenguaje hecho para poder construir otros lenguajes. Es una versión simple de SGML y se encuadra dentro de los lenguajes estructurales.

Los ficheros de texto que utilicen XML para su construcción poseen dos estructuras, una lógica y otra física. Tiene que tener una estructura física correcta, y seguir una estructura semántica especificada por un DTD o XML Schema.

JavaServer Pages [20]

JavaServer Pages (JSP) es una tecnología Java que permite generar contenido dinámico para Web, en forma de documentos HTML, XML o de otro tipo. Esta tecnología es un desarrollo de la compañía Sun Microsystems.

JSP puede considerarse como una manera alternativa, y simplificada, de construir servlets. Cada versión de la especificación de JSP está fuertemente vinculada a una versión en particular de la especificación de servlets.

El funcionamiento general de la tecnología JSP es que el Servidor de Aplicaciones interpreta el código contenido en la página JSP para construir el código Java del servlet

a generar. Este servlet será el que genere el documento (típicamente HTML) que se presentará en la pantalla del Navegador del usuario.

Apache Struts [14]

Apache Struts es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MC bajo la plataforma Java EE (Java Enterprise Edition).

Struts permite reducir el tiempo de desarrollo. Gracias a ser software libre y su compatibilidad con todas las plataformas en las que Java Enterprise esté disponible, lo convierten en una herramienta altamente utilizada.

Struts utiliza y extiende la API Java Servlet para que los programadores utilicen el patrón de arquitectura del software MC. El objetivo de Struts es separar el modelo (lógica de la aplicación que interactúa con una base de datos) de la vista (página HTML presentada al cliente) y el controlador. Struts proporciona el controlador (un servlet conocido como *ActionServlet*) y facilita la escritura de las plantillas para la vista o capa de presentación. El programador de aplicaciones Web, es responsable de escribir el código del modelo, y para crear un archivo de configuración *struts-config.xml* que une modelo, la vista y el controlador.

Framework Tiles [16]

Tiles es un framework de plantillas para simplificar el desarrollo de aplicaciones Web, interfaz de usuario.

Tiles permite a los programadores definir fragmentos de página que se puede montar en una página completa en tiempo de ejecución. Estos fragmentos se pueden utilizar simples para reducir la duplicación de los elementos de la página común, o integrados en otras piezas para desarrollar una serie de plantillas reutilizables. Estas plantillas agilizan el desarrollo de una aplicación completa.

Apache Tomcat [15]

Apache Tomcat es un servidor Web con soporte de servlets y JSPs, que funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en *Apache Software Foundation*. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems.

Tomcat es mantenido y desarrollado por miembros de *Apache Software Foundation* y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en *Apache Software Licence*.

JDBC [19]

JDBC [Java Database Connectivity] es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

El API JDBC se presenta como una colección de interfaces Java y métodos de gestión de manejadores de conexión hacia cada modelo específico de base de datos. Para utilizar una base de datos particular, el usuario ejecuta su programa junto con la biblioteca de conexión apropiada al modelo de su base de datos, y accede a ella estableciendo una conexión, para ello provee el localizador a la base de datos y los parámetros de conexión específicos.

MySql [33]

MySql es un sistema de gestión de bases de datos relacional, multihilo y multiusuario. *MySQL AB* desarrolla MySQL como software libre en un esquema de licencia dual. Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso.

MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones Web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

SQL [26]

SQL [Structured Query Language] es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas.

Es un lenguaje declarativo de *alto nivel*, que gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros, y no a registros individuales, permite una alta productividad en codificación y orientación a objetos.

JavaScript [25]

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en el lado del cliente (client-side), implementado como parte de un navegador Web permitiendo mejoras en la interfaz de usuario y páginas Web dinámicas, aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS).

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas Web.

iFrame [24]

Iframe es un elemento HTML que permite insertar o incrustar un documento HTML dentro de un documento HTML principal. Actualmente la etiqueta Iframe es

aceptada por la W3C como un elemento estándar.

Los iFrames admiten diversos atributos que permite una visualización de la página mucho más atractiva. Además, el uso de iFrames sirvió como inspiración a Ajax, ya que cada parte de un documento html es independiente.

DWR [22]

DWR [Direct Web Remoting] es una librería de código abierto escrita en Java que permite escribir aplicaciones Web con Ajax.

DWR permite que el desarrollador Web utilice el código Java desde JavaScript como si fuera local en el navegador Web, mientras que en realidad el código Java se ejecuta en el servidor Web. Por razones de seguridad el desarrollador Web debe configurar con exactitud qué clases de Java son seguras para la exportación (*web.xml* o *dwr.xml*).

DOM [28]

DOM [Document Object Model] es esencialmente un API que proporciona un conjunto estándar de objetos para representar documentos XHTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos. A través del DOM, los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos XHTML y XML, que es para lo que se diseñó principalmente.

El responsable de DOM es World Wide Web Consortium (W3C) [23].

3.5. PRUEBAS

Las pruebas realizadas, tanto al editor como a la aplicación Web, y a la generación y visualización de documentos, han tenido como misión comprobar que se han cumplido los objetivos iniciales especificados en los requisitos como los que han ido surgiendo a lo largo del desarrollo.

En cuanto al editor y a la aplicación Web, las pruebas han sido realizadas de forma acumulativa, de forma que para cada nueva funcionalidad implementada se realizaban las pruebas pertinentes, consiguiendo de esta forma solucionar pequeños problemas de análisis y diseño cuando era necesario.

Y, en cuanto a la generación y, posteriormente, visualización de documentos offline/online, las pruebas fueron realizadas primero en offline. Y, una vez hecho esto, se empezó con el desarrollo de la parte online, implementando la misma lógica. Y, una vez implementada, se volvieron a hacer el mismo tipo de pruebas.

Por las limitaciones de la memoria no se describen todas las pruebas realizadas sino sólo las más importantes.

Editor

- Prueba del Modelo de datos

Para el modelo de datos se comprobó que recogía todas las características de todos los posibles documentos que se puedan construir. Para ello, se construyeron distintos modelos de documentos. Además, para los componentes multimedia y evaluador se añadieron todos los posibles tipos.

- Prueba de la Interfaz gráfica

Esta prueba tiene como misión medir la usabilidad de la aplicación, y fue llevada a cabo en todo momento con cada una de las pruebas, ya que para la realización de las mismas, se tuvo que utilizar la interfaz gráfica.

- Pruebas de funcionamiento:

- **Creación de un documento y navegación**

Esta prueba pretende comprobar la correcta creación de un documento y de su navegación. Para ello, se tienen que crear varios documentos, con sus diagramas de navegación, donde, en cada uno de ellos, se añadirán pasos al diagrama y reglas (arcos), entre los pasos, pudiéndose validar, en todo momento, dichos grafos.

- **Edición de un paso e inclusión de componentes**

Esta prueba pretende comprobar la correcta edición de un paso, con la inclusión de diferentes componentes. Para ello, una vez que se haya añadido un paso al diagrama de navegación, se hará *doble click* sobre él, y se podrá editar. Añadiéndose los componentes narrativos y multimedia que se quieran, pero sólo pudiendo añadir un componente evaluador, por cada paso.

- **Edición de componentes narrativos**

En esta prueba se pretende comprobar también la correcta edición de los componentes narrativos, donde se pueden añadir todos los párrafos que se quieran y elegir una *Clase CSS* específica para ellos, de tal manera que se muestre con la fuente, tamaño, color, etc. que se elija.

- **Edición de componentes multimedia**

Para la prueba de edición de componentes multimedia, se pretenden añadir los distintos tipos que existen, y su correcto funcionamiento. Pudiendo elegir diferentes características, dependiendo de la elección del tipo (ancho, alto, rutaFichero, autoPlay, etc.).

- **Edición de componentes evaluadores**

Y, por último, la prueba de edición de componentes evaluadores consiste en añadir preguntas de los diferentes tipos (test, un valor y rellenar huecos), rellenando la información necesaria para cada una de ellas, y sus respuestas.

Repositorio

- **Prueba de la interfaz gráfica:** administrador, docente y alumno.

Esta prueba tiene como misión medir la usabilidad de la aplicación, y fue llevada a cabo en todo momento con cada una de las pruebas, ya que para la realización de las mismas, se tuvo que utilizar la interfaz gráfica.

- **Docente:** Inclusión de un nuevo documento, generación y asociación de documentos y grupos.

Esta prueba es la base de la aplicación Web, para ella el docente, previamente, ha tenido que crear el documento Xml con una estructura concreta. Una vez hecho esto, el docente selecciona de su PC el documento comprimido .zip que contiene toda la información necesaria, y lo añade a la aplicación Web. Se comprobó que el documento era añadido, junto con otros archivos multimedia, al servidor Web, y que por tanto, podía ser utilizado, a partir de ese momento, para su generación offline y online.

Una vez realizada la generación offline, se comprobó que en el servidor Web aparecía un archivo comprimido, el cual podría ser más tarde descargado por el alumno, que contenía todo lo necesario para su visualización.

Y, una vez realizada la generación online, se comprobó que en la base de datos aparecían fragmentos de los pasos que posteriormente se utilizarán para la visualización del documento online.

- **Alumno:** Visualización de documentos online.

En esta prueba se pretende completar lo anteriormente descrito. El alumno, una vez seleccionado el documento a descargar, sólo pudiéndose descargar aquellos documentos de su asignatura, puede ver el documento offline desde su navegador Web, sin necesidad de conexión a Internet.

- **Alumno:** Descarga y visualización de documentos offline.

En esta prueba también se pretende completar lo anteriormente descrito en la sección del Docente. El alumno, una vez seleccionado el documento a visualizar

online, sólo pudiéndose ver aquellos documentos de su asignatura, puede visualizar dicho documento, siendo necesaria la conexión a Internet, y lo respondido en él se guarda en nuestra base de datos. Comprobándose que el estado de conocimiento del alumno se ha guardado correctamente.

■ **Administrador:** Inclusión de nuevos usuarios.

Esta prueba pretende comprobar la correcta inclusión de usuarios tanto en la aplicación Web como en la base de datos. El administrador será el único que podrá añadir nuevos usuarios, donde, una vez realizada la acción, se comprobará que:

- En la aplicación Web, esos nuevos usuarios pueden acceder al sistema, en el rol en el que han sido creados, y no pueden acceder a otras secciones del sistema, porque no tienen permisos.
- En la base de datos, esos nuevos usuarios han sido dados de alta de manera correcta.

Visualización de documentos

■ **Prueba del Modelo de Datos.**

Para el modelo de datos se comprobó que recogía todos los tipos de componentes, y dentro de cada componente sus diferentes posibilidades. Además, se comprobó como, dependiendo del tipo de navegación, se creaban los pasos destino, dependiendo de las reglas de reproducción del documento.

■ **Visualización con navegación secuencial.**

Esta prueba es una de las bases del proyecto, en ella se pretende hacer hincapié en la navegación secuencial. Donde se comprobó que el documento interactivo no varía dependiendo de las respuestas del alumno, el orden es siempre el mismo.

■ **Visualización con navegación adaptativa.**

Esta prueba también es una de las bases del proyecto, en ella se pretende hacer

hincapié en la navegación adaptativa. Donde se comprobó que el documento interactivo varía dependiendo de las respuestas correctas o incorrectas del alumno, adaptándose al conocimiento de este, ayudándole y repitiendo aquellos pasos, cuyos niveles de conocimiento no ha adquirido.

3.5.1. Ejemplo de funcionamiento

En esta sección, se pretende describir un ejemplo de uso del editor, aplicación Web y la visualización de documentos, mediante pantallas y explicaciones.

3.5.1.1. Editor

Para describir el editor, se presenta un ejemplo de creación de un documento, desde cero, con la edición de la navegación entre los diferentes pasos, y la creación y edición del contenido de esos pasos.

1. **Arrancar el *Editor de documentos***. La primera pantalla muestra en el menú: *File*, *Edit*, *Estructuras*, *Generar*, *Exportar*, *Window* y *Help*.
 - *File* con submenú *New*, que permite crear documentos (Documento Diagram), pasos (Paso Diagram) y modelar estructuras (Estructuras Model).
 - *Estructuras* con dos submenús: *Estructuras* y *Estilo Css*. Donde *Estructuras* permite editar el fichero de estructuras (para los componentes multimedia), y *Estilo Css*, para editar el fichero de estilos.
 - *Generar* con submenú *Generar offline*, que permite la generación offline del documento, para su visualización Web.

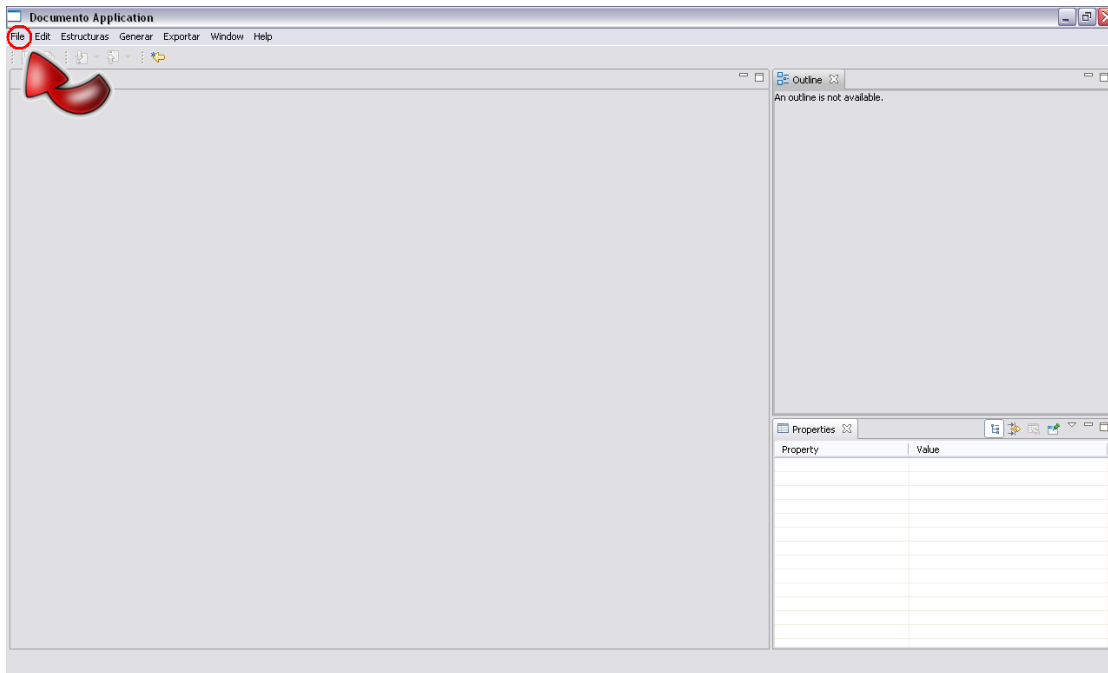


Figura 3.106: Primera pantalla de la aplicación.

2. **Crear un nuevo documento y diagrama de navegación.** El primer y segundo diálogo para crear el documento y diagrama de navegación. En el primero, se puede cambiar el nombre del diagrama de navegación, y en el segundo, se puede cambiar el nombre del documento.

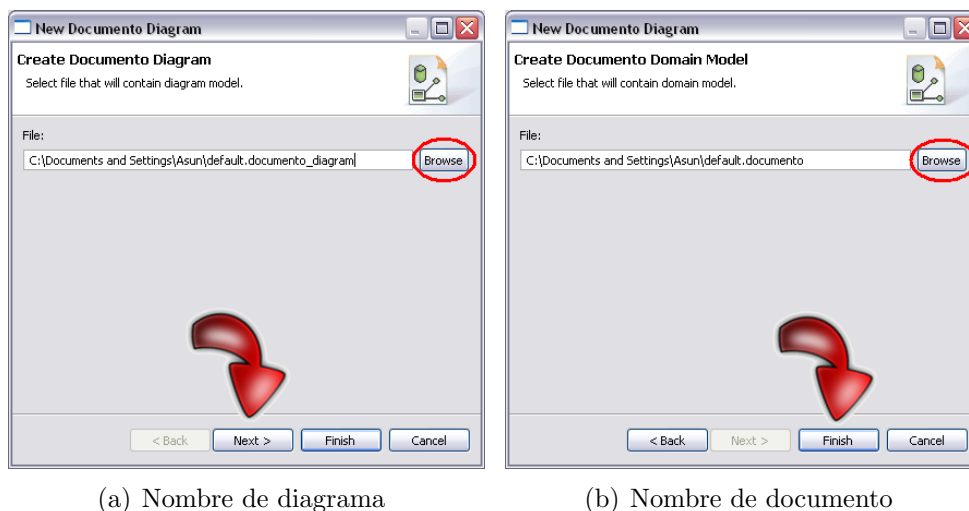


Figura 3.107: Diálogo para crear un nuevo documento y diagrama de navegación.

3. **Editar documento de estructuras.** Se puede editar el documento de estructuras, para los componentes multimedia, desde el menú *Estructuras*, con submenú también *Estructuras*.

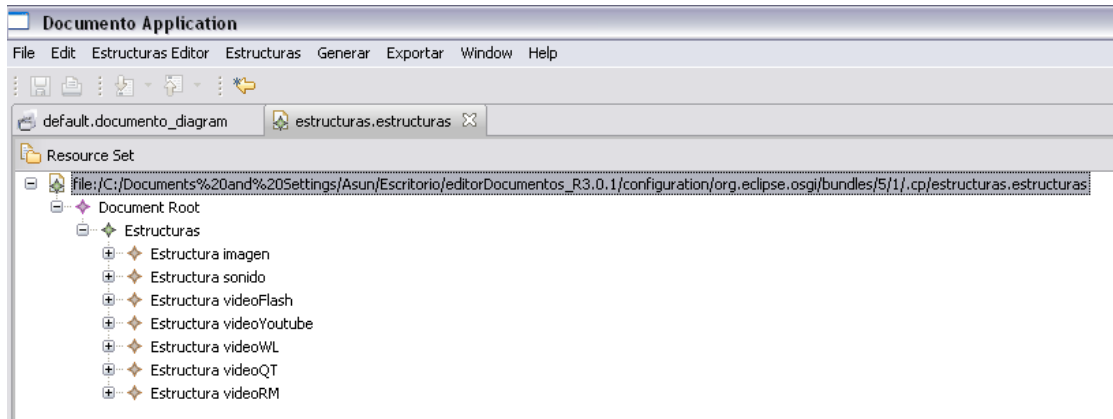


Figura 3.108: Diálogo para editar documento de estructuras.

4. **Editar documento de estilo.** Se puede editar el documento del estilo, para cambiar fuentes, tamaño, estilo, etc.

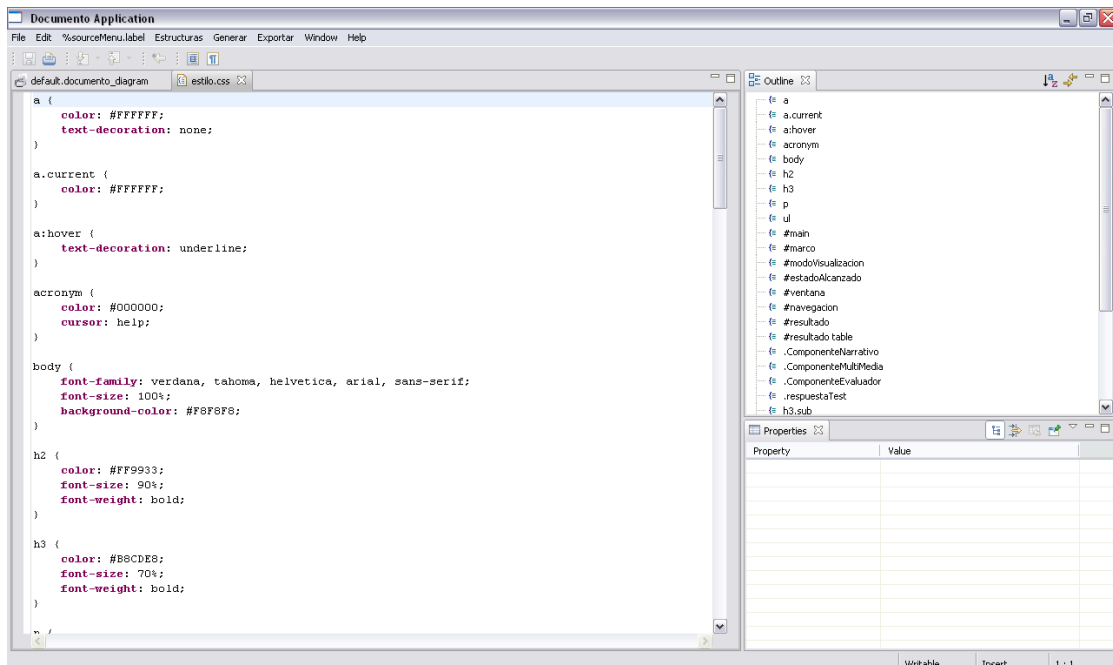


Figura 3.109: Diálogo para editar documento de estilo.

Vista: diagrama de navegación

5. **Introducir pasos y reglas.** Se añaden pasos, y después reglas que unen estos pasos, creando el grafo de navegación, que seguirá el documento.

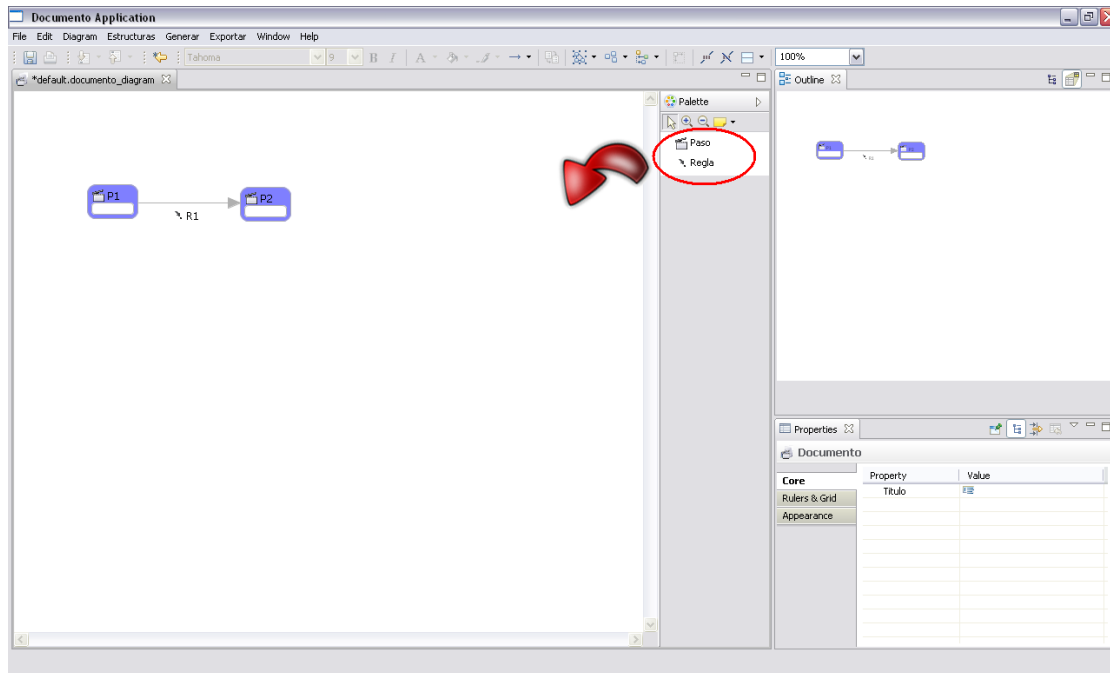
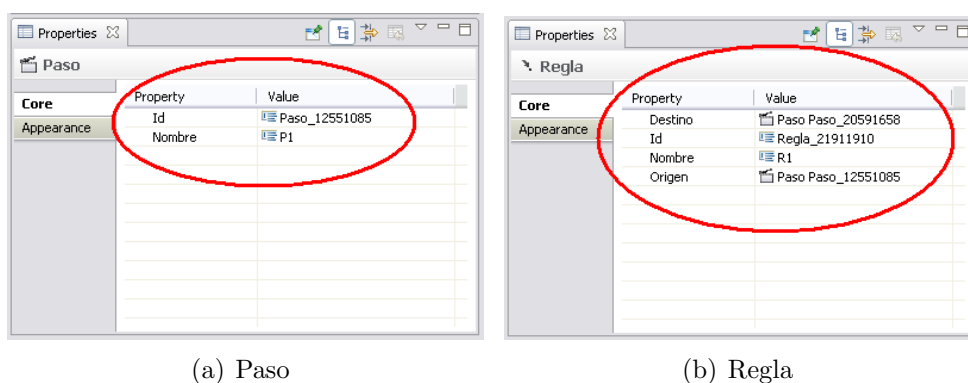


Figura 3.110: Diálogo para crear un nuevo documento y diagrama de navegación.

6. **Editar pasos y reglas.** Se pueden editar las propiedades de los pasos y reglas.



(a) Paso

(b) Regla

Figura 3.111: Diálogo para editar propiedades de los pasos y reglas de navegación.

7. **Validar navegación.** Se puede validar el diagrama de navegación en cualquier momento, que devolverá un listado de errores.

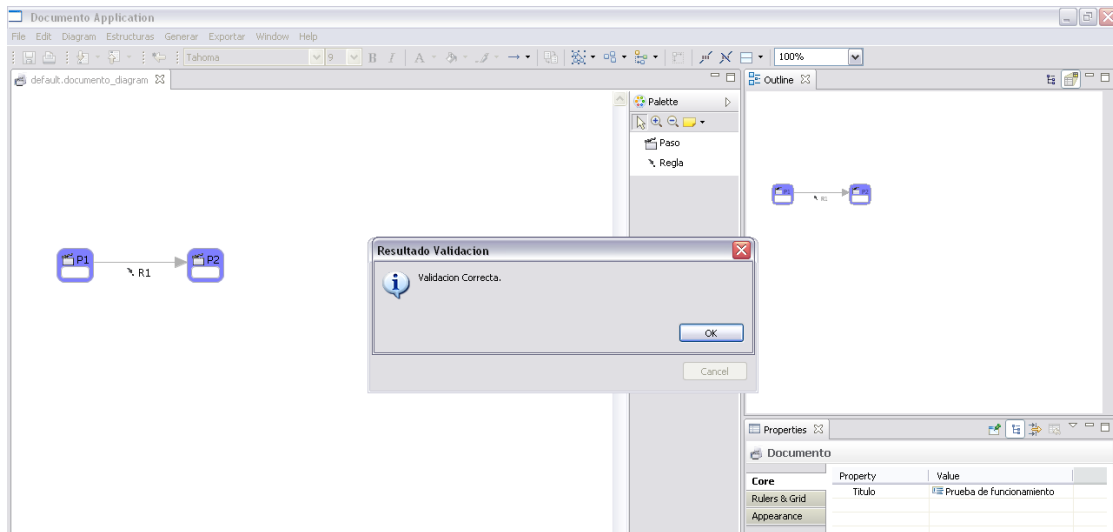


Figura 3.112: Diálogo para validar navegación.

Vista: diagrama de edición de pasos

8. **Añadir componentes.** Se añaden componentes narrativos (verde), multimedia (azul) y evaluadores (rojo), al paso del documento.

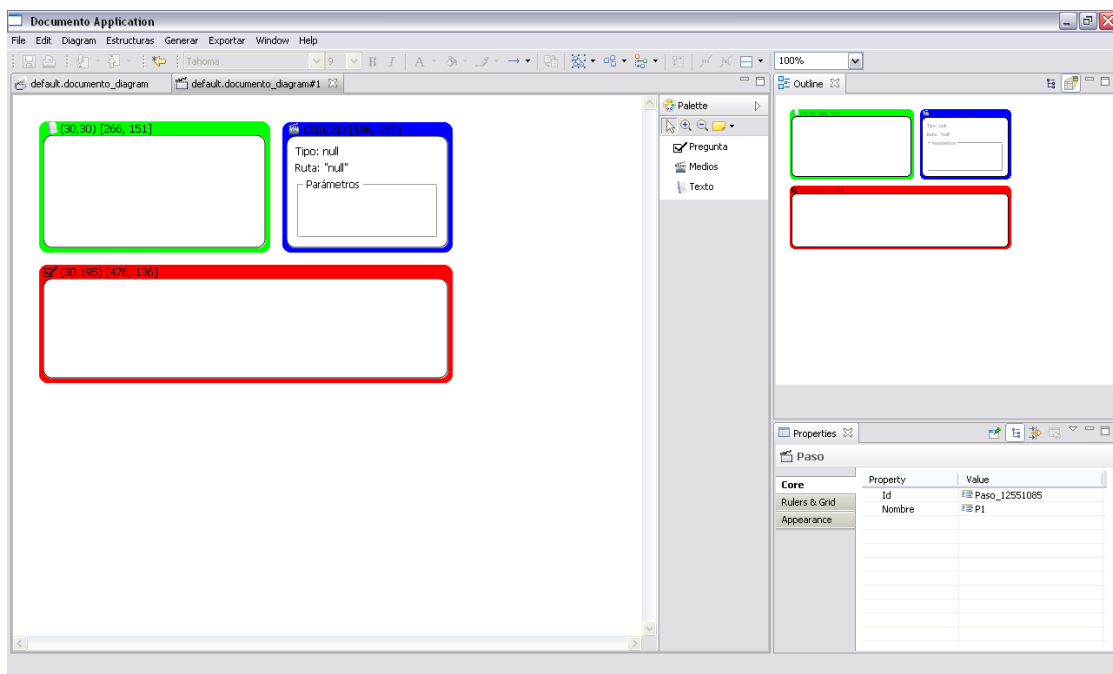
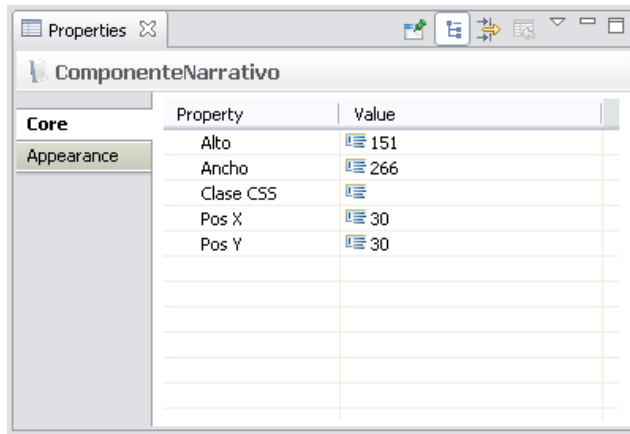
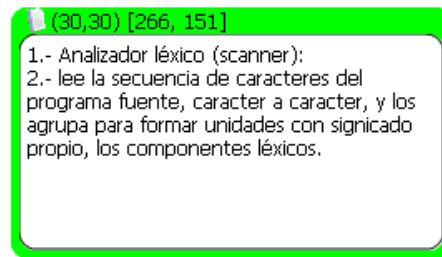


Figura 3.113: Diálogo para añadir componentes: narrativo, multimedia y evaluador.

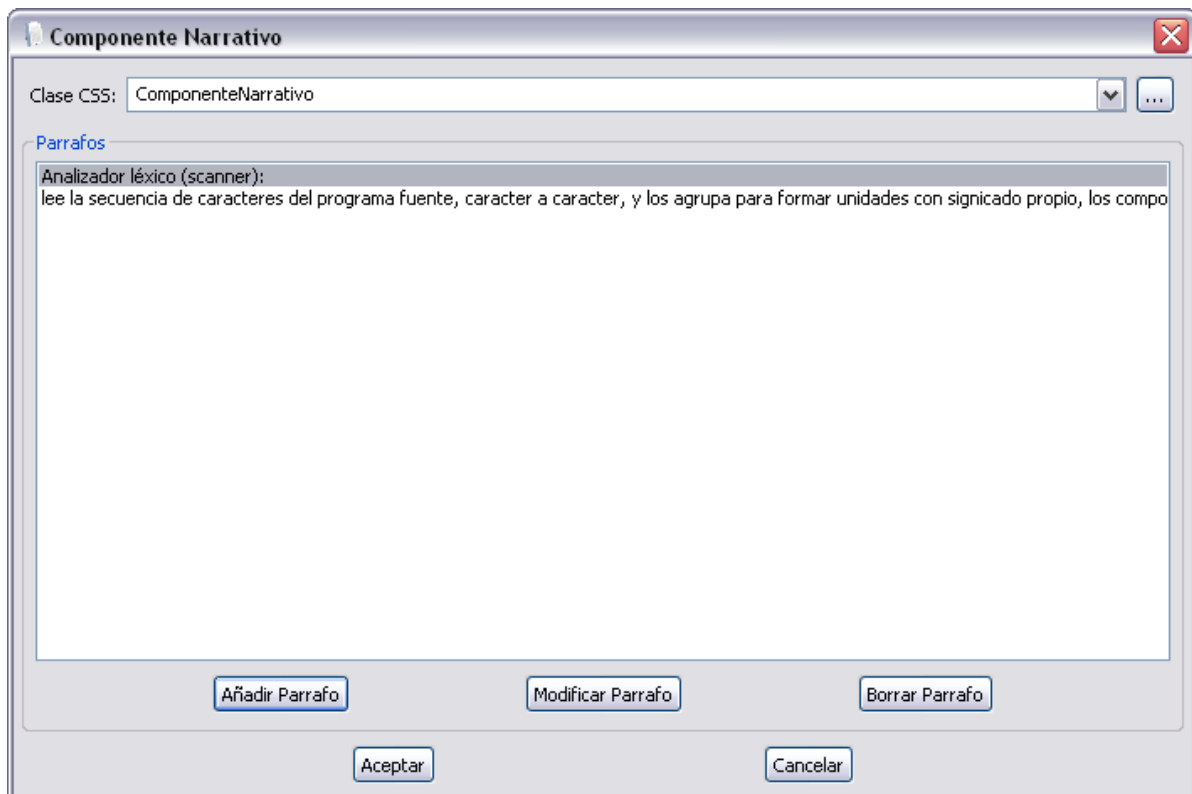
9. **Editar componente narrativo.** Se pueden editar las propiedades y el resto de características (añadir párrafos) de los componentes narrativos.



(a) Propiedades



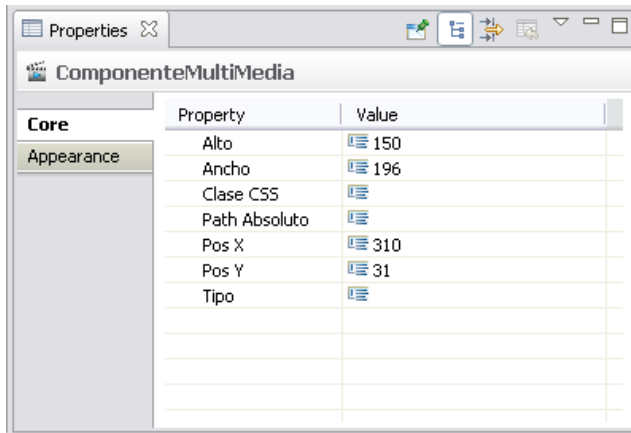
(b) Resultado



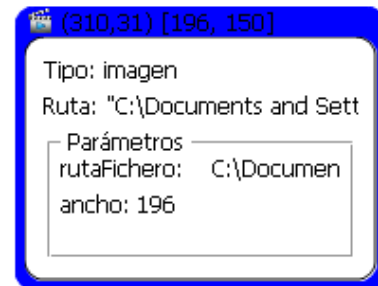
(c) Menú contextual: Modificar

Figura 3.114: Diálogo para editar componente narrativo.

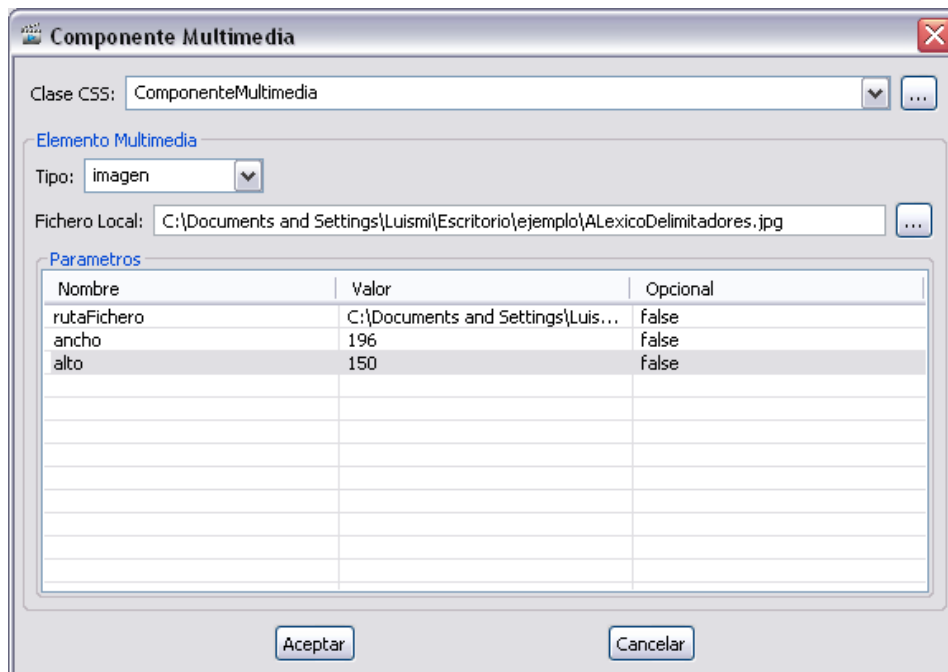
10. **Editar componente multimedia.** Se pueden editar las propiedades y el resto de características (Clase Css, Tipo, Fichero local y parámetros) de los componentes multimedia.



(a) Propiedades



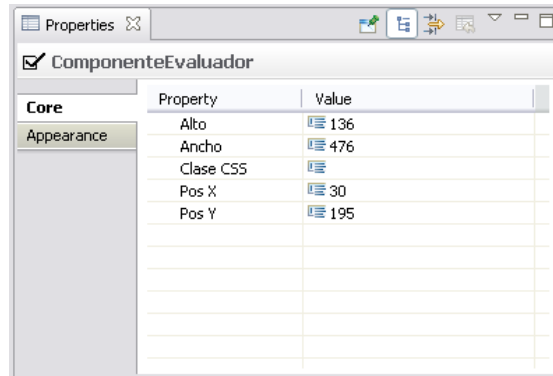
(b) Resultado



(c) Menú contextual: Modificar

Figura 3.115: Diálogo para editar componente multimedia.

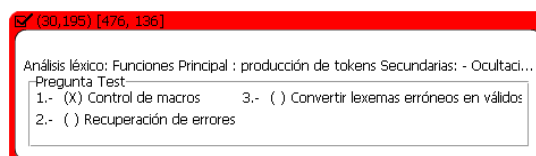
11. **Editar componente evaluador.** Se pueden editar las propiedades y el resto de características (Clase Css, Enunciado, Tipo y Respuestas).



(a) Propiedades



(b) Menú contextual: Modificar



(c) Resultado

Figura 3.116: Diálogo para editar componente evaluador.

12. **Añadir estado de conocimiento al componente evaluador.** Se puede editar el estado de conocimiento del componente evaluador, añadiendo conocimientos con nivel (conocimiento, comprensión, aplicación, análisis, síntesis y evaluación).

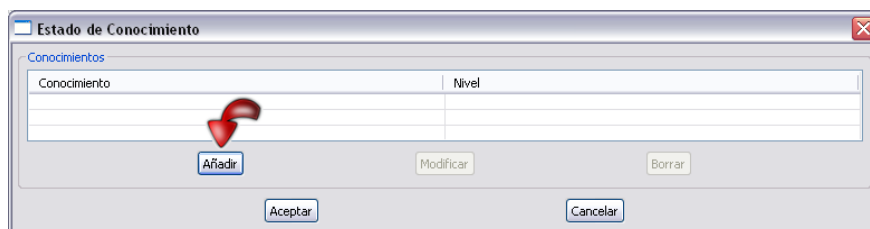


Figura 3.117: Diálogo para editar el estado de conocimiento del componente evaluador.

13. **Exportar Web.** Una vez construidos todos los pasos y su contenido, y navegación, se puede exportar a la Web. Para ello se creará un documento comprimido, que contenga toda la información necesaria para visualizarlo.

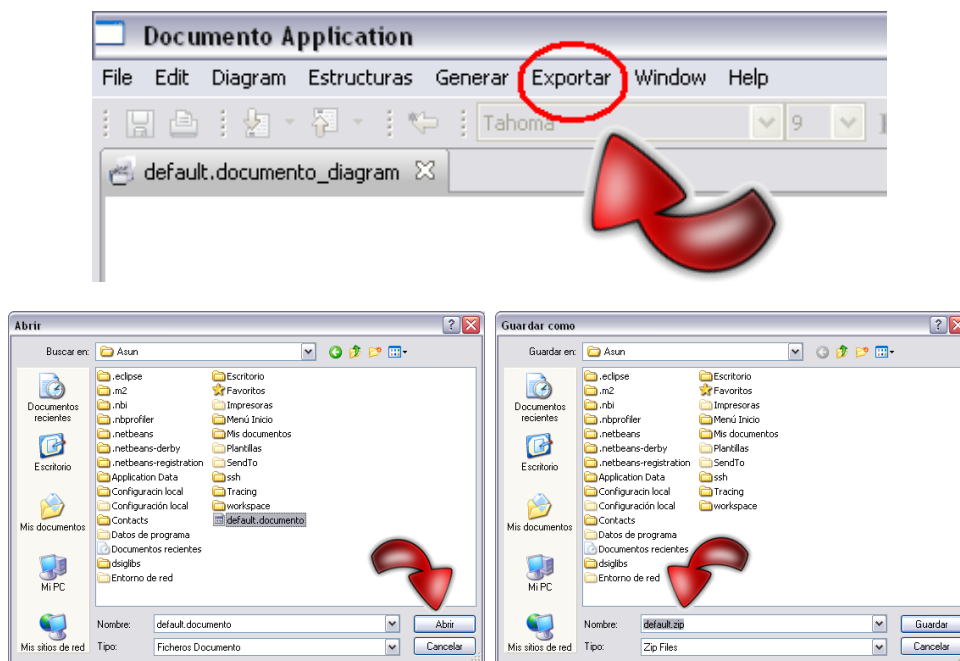


Figura 3.118: Diálogo para exportar a la Web.

14. **Generar documento de visualización Web.** También, una vez construidos todos los pasos y su contenido, y navegación, se puede generar un documento, para su visualización en la Web.

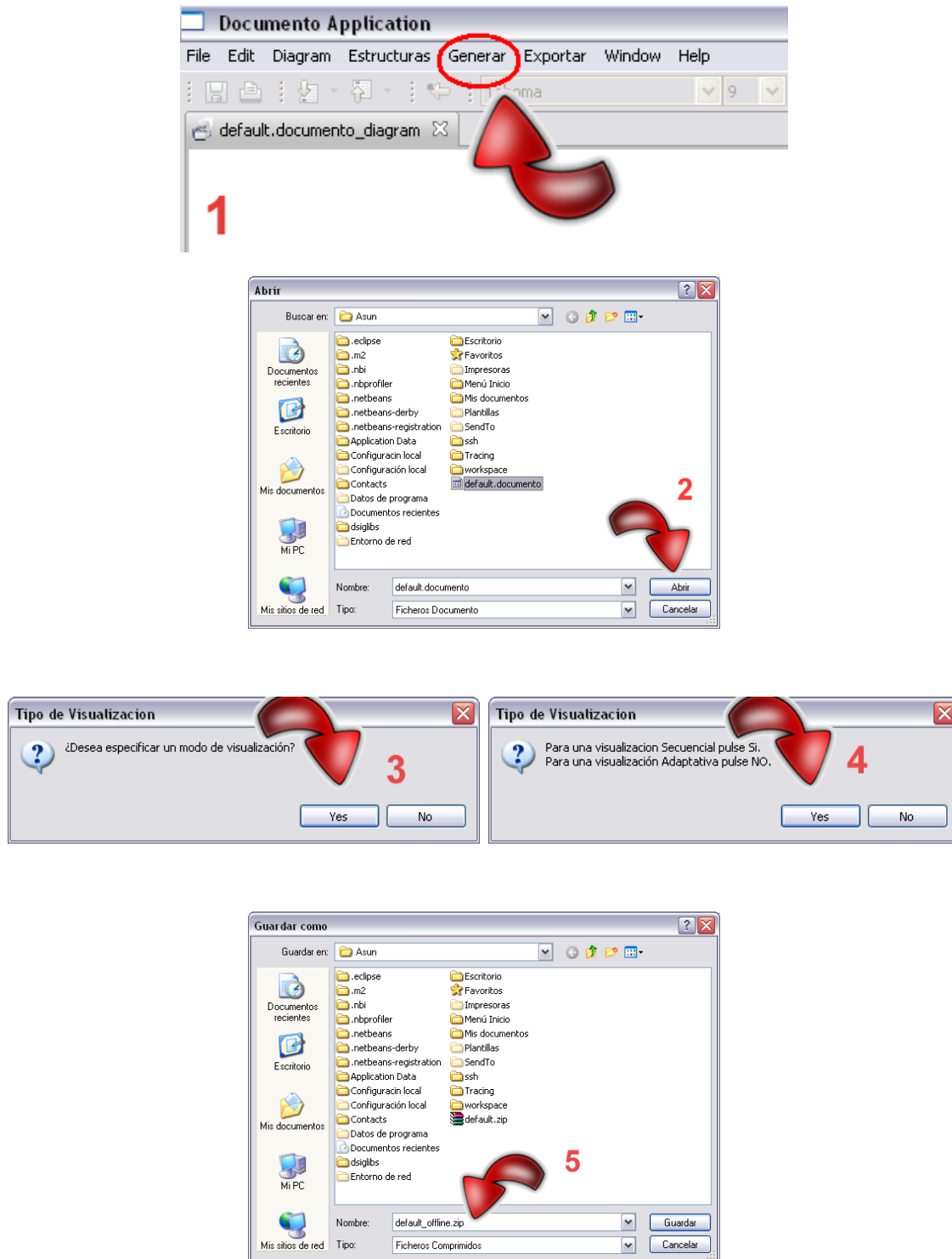


Figura 3.119: Diálogo para generar un documento offline.

3.5.1.2. Repositorio

Para describir la aplicación Web, se presenta un ejemplo donde se añade un fichero comprimido, se generan las presentaciones offline y online. Además, se asocia dicho documento con el/los grupo/s del profesor, pudiendo también asociar el/los grupo/s con alumnos. Finalizando, cuando el alumno se descarga las presentaciones offline o visualiza las presentaciones online.

Docente

1. **Arrancar la aplicación Web:** la primera pantalla muestra el *Acceso al sistema*. Donde el administrador, docente y alumno tendrán que introducir su usuario y contraseña. En el caso del docente y alumno, el usuario podría ser la primera parte de su correo electrónico (@urjc.es o @alumnos.urjc.es).

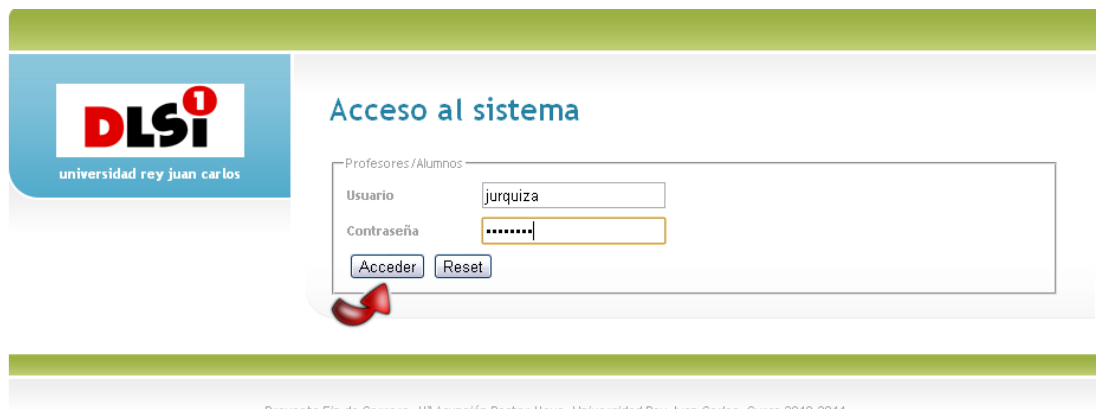


Figura 3.120: Pantalla de acceso a la aplicación Web

2. **Gestión de documentos:** la primera vez que el docente entre verá que no tiene documentos, pudiendo añadir el primero al pulsar “Seleccionar archivo”. Seleccionará el archivo comprimido .zip y se añadirá a la aplicación Web dicho documento, pulsando “Subir documento”.



Figura 3.121: Pantalla: Gestión Documentos

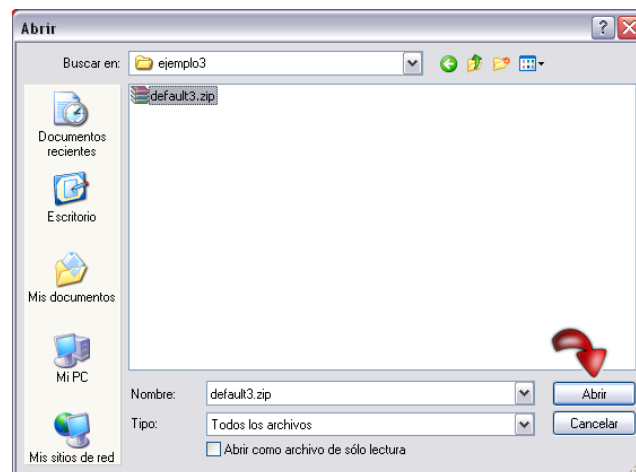


Figura 3.122: Selección del documento .zip



Figura 3.123: Subir documento al Servidor

3. **Generación offline y online:** El profesor seleccionará el documento añadido, y pulsará en su generación offline (secuencial/adaptativa) o en su generación online (secuencial/adaptativa).

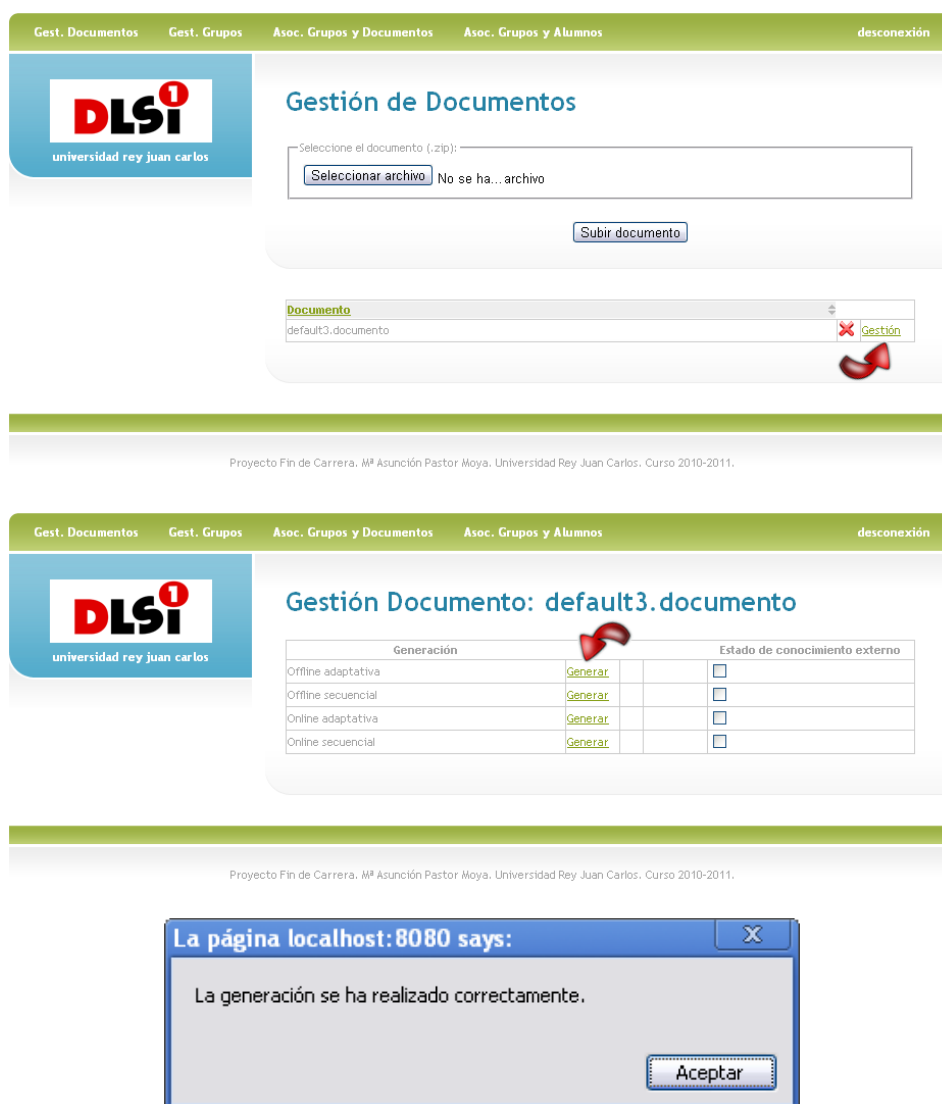


Figura 3.124: Generar documento offline u online

4. **Gestión de grupos:** una vez añadido el documento, tendrá que crear el grupo que asociará con dicho documento. La primera vez que el docente entre verá que no tiene grupos, pudiendo crear el primero escribiendo su nombre y pulsando "Crear grupo". Añadiéndose dicho grupo a la aplicación Web y a la base de datos.



Figura 3.125: Pantalla: Gestión Grupos



Figura 3.126: Crear grupo

5. **Asociación de grupos y documentos:** una vez creado/s el/los grupo/s, el profesor tendrá que asociar el documento añadido a dicho grupo. La primera vez que el docente entre verá que no hay ninguna asociación, pudiendo seleccionar el documento y grupo/s que quiera asociar y pulsar “Asociar”. Añadiéndose dicha asociación a la aplicación Web y a la base de datos.

Gest. Documentos Gest. Grupos Asoc. Grupos y Documentos Asoc. Grupos y Alumnos desconexión

DLSI
universidad rey juan carlos

Asociación de grupos y documentos

Documentos: my.documento

Grupos: Procesadores de lenguajes

Nothing found to display.

1, 2, 3

Proyecto Fin de Carrera. M^a Asunción Pastor Moya. Universidad Rey Juan Carlos. Curso 2010-2011.

Figura 3.127: Pantalla: Asoc. Grupos y Documentos

Gest. Documentos Gest. Grupos Asoc. Grupos y Documentos Asoc. Grupos y Alumnos desconexión

DLSI
universidad rey juan carlos

Asociación de grupos y documentos

Documentos: my.documento

Grupos: Procesadores de lenguajes

Grupo	Documento
Procesadores de lenguajes	my.documento

Proyecto Fin de Carrera. M^a Asunción Pastor Moya. Universidad Rey Juan Carlos. Curso 2010-2011.

Figura 3.128: Asociar documento a grupo/s

- Asociación de grupos y alumnos:** ahora, el profesor tendrá que asociar el documento añadido a los alumnos que desee. Seleccionará el documento y alumno/s que quiera asociar y pulsar “Asociar”. Añadiéndose dichas asociaciones a la aplicación Web y a la base de datos.



Proyecto Fin de Carrera. M^a Asunción Pastor Moya. Universidad Rey Juan Carlos. Curso 2010-2011.

Figura 3.129: Pantalla: Asoc. Grupos y Alumnos



Proyecto Fin de Carrera. M^a Asunción Pastor Moya. Universidad Rey Juan Carlos. Curso 2010-2011.

Figura 3.130: Asociar documento a alumno/s

Alumno

1. **Visualización de documentos:** al igual que el profesor, al arrancar la aplicación, el alumno tendrá que escribir su usuario y contraseña. Una vez hecho esto, si

el docente ya ha creado y generado los documentos offline y online, podrá pulsar en “Path offline” y descargarse el documento .zip, o bien pulsar en “Path online” y visualizarlo en la Web.



Figura 3.131: Pantalla: Ver documentos

Administrador

1. **Gestión de usuarios:** al igual que el profesor y alumno, al arrancar la aplicación, el administrador tendrá que escribir su usuario (admin) y contraseña. Una vez hecho esto, podrá crear nuevos usuarios, eliminar existentes y editar algunas de sus propiedades.



Figura 3.132: Pantalla: Gestión Usuarios

3.5.1.3. Visualización de documentos

Para describir la visualización de documentos, se presenta un ejemplo de presentación interactiva, donde se pueden observar los diferentes componentes: narrativo (texto), multimedia (imágenes, videos y sonido) y evaluador (preguntas).

1. Primer paso:

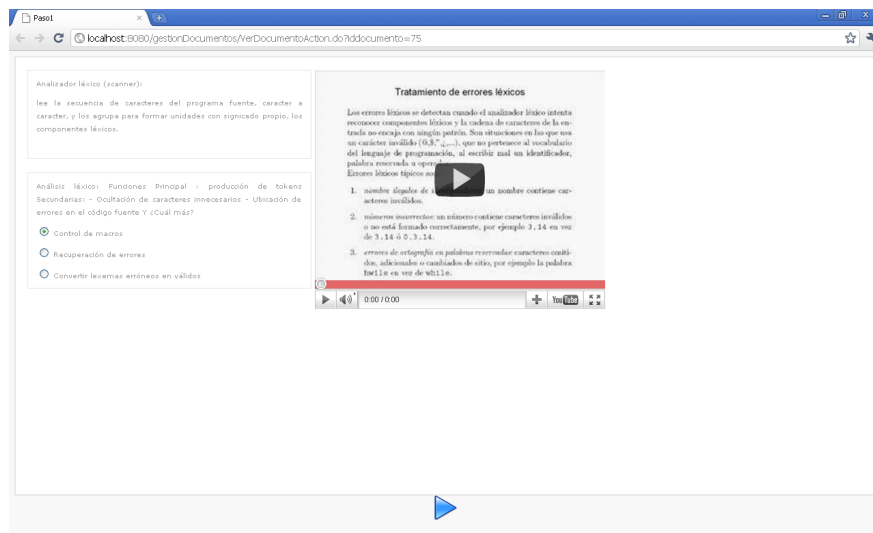


Figura 3.133: Paso 1: Componente narrativo, multimedia (video YouTube) y evaluador (test)

2. Segundo paso:

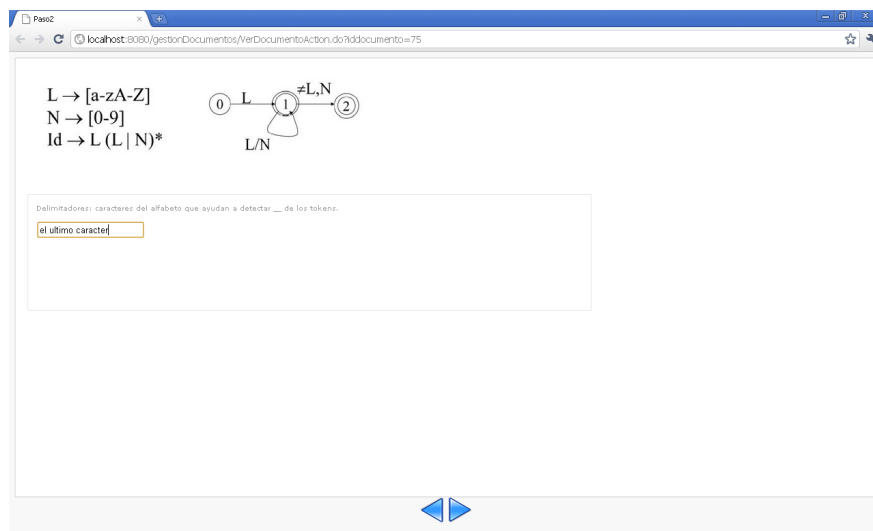


Figura 3.134: Paso 2: Componente multimedia (imagen) y evaluador (valor único)

3. Tercer paso:

El A.L. debe traducir los lexemas a tokens. ★
traductor finito = A.F.D. + *acciones semánticas*.

Las acciones más comunes son:

- Leer siguiente carácter, concatenar caracteres que forman el lexema, calcular valor numérico, entregar tokens al A.S.....
- Leer caracteres del lexema, calcular el siguiente carácter, diferenciar entre carácter numérico y alfanumérico,...
- Leer palabras reservadas, buscar caracteres repetidos, retroceder en el buffer de entrada, entregar tokens al A.S.....

Figura 3.135: Paso 3: Componente multimedia (imagen) y evaluador (test)

4. Cuarto paso:

Expresiones regulares ★

Definición de lenguajes sobre alfabetos.

- "ε".
- "a" siempre que "a" ∈ alfabeto.
- Si r y s son cadenas que pertenecen respectivamente a los lenguajes $L(r)$ y $L(s)$, son expresiones regulares:
 - a. $r|s \equiv L(r) \cup L(s)$
 - b. $rs \equiv L(r)L(s)$
 - c. $r^* \equiv (L(r))^*$

Expresiones regulares y A.F.D.

Alfabeto = de símbolos

Cadena sobre un alfabeto = de de ese alfabeto

Lenguaje = de cadenas sobre un alfabeto

Figura 3.136: Paso 4: Componentes multimedia (Flash e imagen) y evaluador (valor: rellenar huecos)

5. Quinto paso:

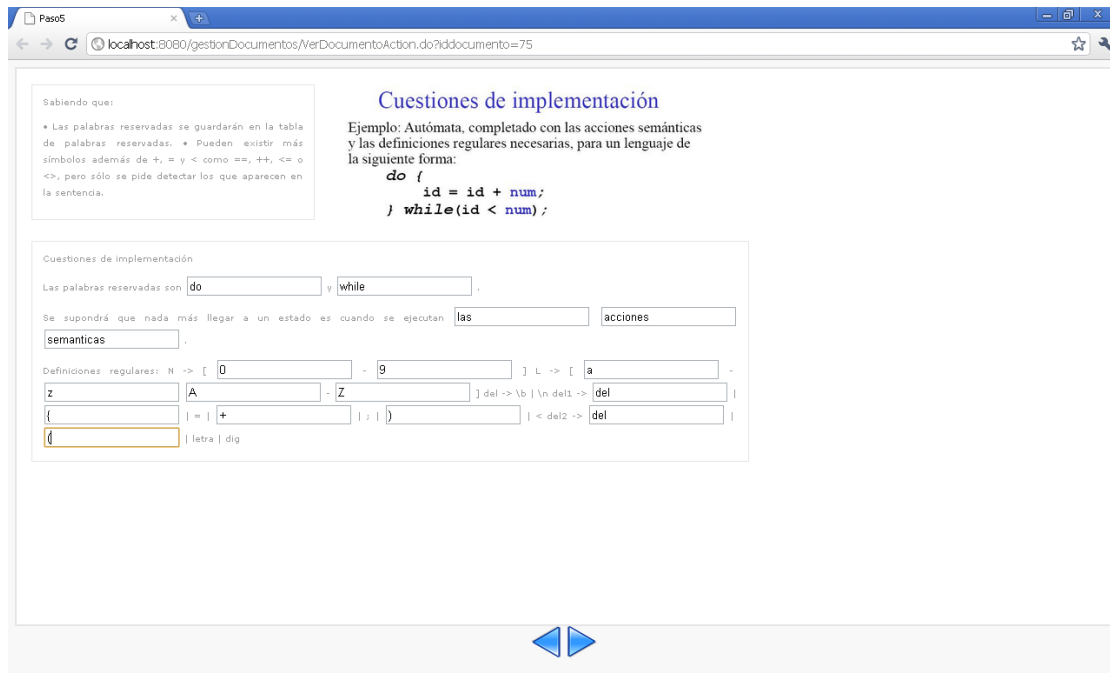


Figura 3.137: Paso 5: Componente narrativo, multimedia (imagen) y evaluador (valor: rellenar huecos)

6. Sexto paso:

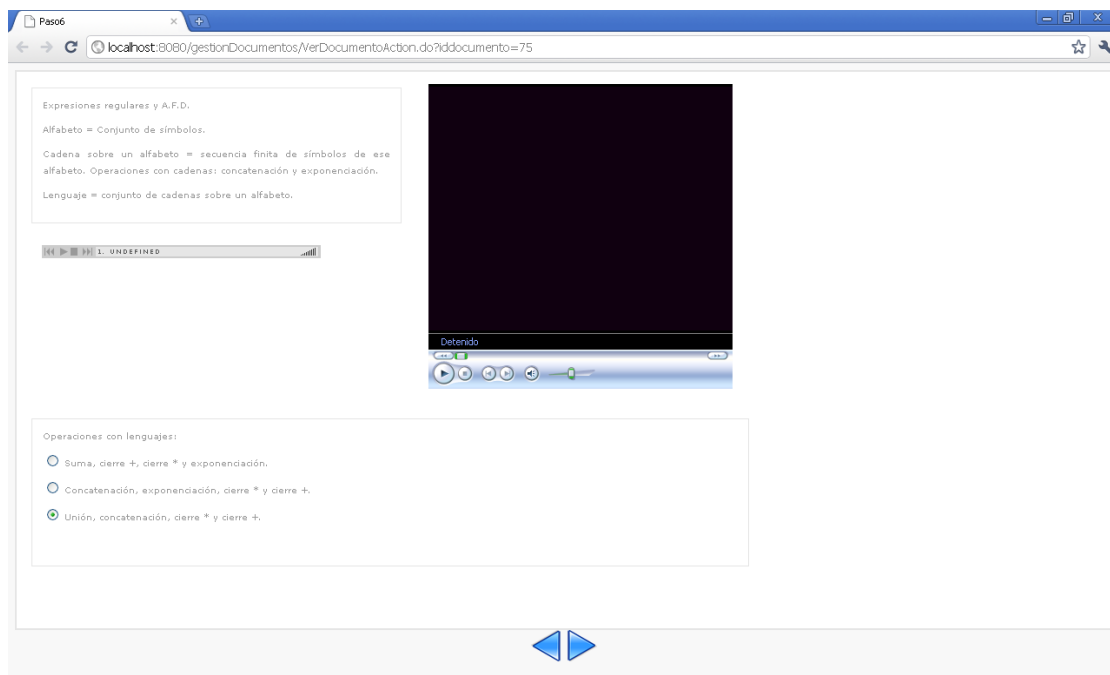


Figura 3.138: Paso 6: Componente narrativo, multimedia (video .avi y sonido) y evaluador (test)

7. Séptimo paso:

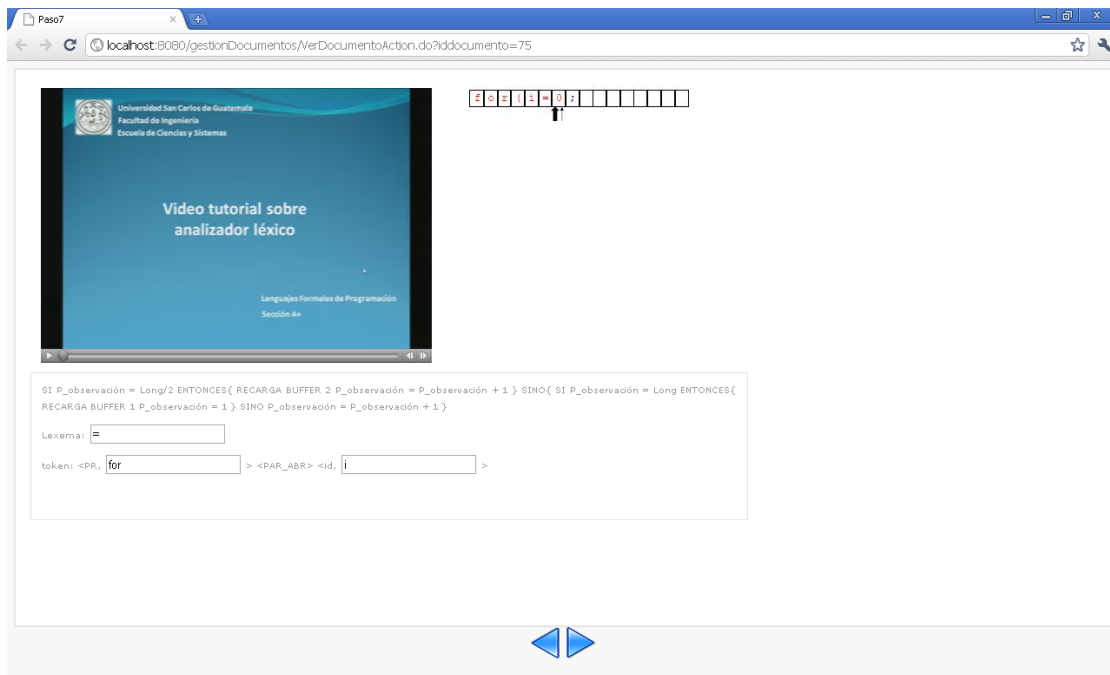


Figura 3.139: Paso 7: Componente multimedia (video .mov e imagen) y evaluador (valor: rellenar huecos)

8. Resultado de la visualización:

Estado de conocimiento Alcanzado:	
CONCEPTO	NIVEL
Lenguajes y Sistemas/Procesadores del lenguaje/Análisis léxico/Expresiones regulares	Análisis
Lenguajes y Sistemas/Procesadores del lenguaje/Análisis léxico/Funciones	Conocimiento
Lenguajes y Sistemas/Procesadores del lenguaje/Análisis léxico/Definición	Conocimiento
Lenguajes y Sistemas/Procesadores del lenguaje/Análisis léxico/Implementación	Análisis
Lenguajes y Sistemas/Procesadores del lenguaje/Análisis léxico/AFD	Análisis

Figura 3.140: Ejemplo: Resultado de Visualización offline/online

CONCLUSIONES

4.1. Evaluación

Editor

El principal objetivo del proyecto, con respecto al *editor de documentos*, ha sido la realización de una aplicación para la creación y edición de documentos formativos, siendo realizado con éxito.

Para la realización de esta aplicación se ha precisado el estudio de las librerías SWT y JFace de Java, además de la profundización en el API de Java. También, para el editor, se ha necesitado el estudio de los frameworks de Eclipse EMF (para el modelo de datos), GEF (para el editor gráfico) y GMF.

La gestión de ficheros comprimidos *.zip* precisó de la familiarización con el API de compresión y empaquetado de Java.

Por último, para solventar la edición de ficheros de estilo *.css*, hubo que incorporar a la aplicación un plug-in de Eclipse para su edición. Tendiéndose que adaptar, para su correcto funcionamiento, a dicha aplicación .

Aplicación Web y Generación y visualización de documentos

En primer lugar, se va a destacar la realización correcta de todos los objetivos planteados inicialmente y cómo, a medida que se avanzaba en el desarrollo, han ido apareciendo nuevos objetivos, los cuáles también han sido cumplidos satisfactoriamente.

El objetivo principal fue construir documentos interactivos a partir de archivos Xml. Desde un principio, se pensó en la necesidad de albergar estos documentos en la Web. Pero fue después de realizar una primera parte, generación de documentos offline, cuando se vio realmente esta necesidad y se empezó con la aplicación Web.

Para la realización de la aplicación Web se ha precisado de la profundización en el lenguaje JSP, además del estudio del framework Struts y Tiles.

La generación y visualización de documentos offline y online también ha supuesto un esfuerzo, debido a la cantidad de código en JavaScript que se ha necesitado, sobre todo en la parte offline. Además, se ha tenido que hacer un estudio de DWR [18], para comunicar las respuestas del cliente con Java. De esta forma, en lugar de devolver una página completamente nueva (generada en el servidor) sencillamente, se actualiza (en el cliente) el código HTML de la página actual. En la parte offline, esto se ha resuelto gracias al uso de Iframe.

La conclusión final es, por un lado, la aplicación Web ha supuesto un esfuerzo relativamente pequeño y ha permitido la gestión de los documentos, grupos y alumnos. De tal forma, que los docentes puedan gestionar sus recursos y, a su vez, proporcionar a los alumnos documentos formativos que podrán descargarse en cualquier momento y desde cualquier lugar del Mundo.

Por otro lado, la generación y visualización de documentos offline y online, siendo la parte más importante del proyecto, suponiendo un esfuerzo algo mayor, pero que

proporcionará al alumno una forma entretenida e interactiva de aprender, y le ayudará también a afianzar sus conocimientos. Además, el alumno, al igual que el profesor, podrá ver sus avances y esto le motivará a seguir adelante.

4.2. Trabajos futuros

Editor

Una vez finalizado este proyecto, y tras conocer cual ha sido el alcance del mismo, surgen líneas futuras de ampliación a través de las cuales se puede mejorar el sistema construido, y que harán más agradable y útil su uso, al docente. A continuación se enumeran algunas de ellas:

- Mejorar la interfaz de usuario: más fácil de usar, accesible, etc.
- Incorporar la funcionalidad de poder copiar y pegar componentes, de forma que cada componente sea único.
- Y, añadir nuevos tipos de componentes, añadiéndose así mayor funcionalidad al documento final.

Aplicación Web y Generación y visualización de documentos

Tras la realización del proyecto y haber construido tanto la aplicación Web como documentos interactivos, se pueden observar varios trabajos futuros. Por un lado, se podría aumentar la funcionalidad de la aplicación Web o mejorar algunos aspectos. Por otro lado, a los documentos se les podrían añadir nuevas funcionalidades.

La primera mejora en la aplicación Web, que se ha dejado preparada para sólo implementar, sería el poder tomar el estado de conocimiento del alumno de otras bases de datos, incluso de otros servidores Web, y a partir del cual el alumno podría avanzar en sus conocimientos, sin tener que empezar desde cero.

Además, la aplicación Web podría mejorar el apartado *Asoc. grupos y alumnos*, donde los alumnos aparecieran de una forma más ordenada y fácil de buscar, agrupados por grupos (asignaturas). Esto es, ya que el número de alumnos suele ser mayor. También, podrían añadirse nuevas funcionalidades: el docente podría ver el estado de conocimiento del alumno, descargándose un archivo Excel, o de otro tipo, a su equipo, o verlo desde la aplicación Web, etc.

Por otro lado, las posibles nuevas funcionalidades que se les podrían añadir a los documentos formativos serían:

- Nuevos tipos de componentes a añadir por el docente como: pizarra, visualización de documentos .pdf o de otros tipos, etc.
- Ayuda en los componentes evaluadores, para orientar al alumno en la/s respuesta/s.

Y, por último, una mejora en base de datos podría ser la encriptación de contraseñas y de otros datos importantes, para preservación de la *Protección de datos* de profesores y alumnos.

CONTENIDO DEL CD-ROM

El cd-rom contiene las siguientes carpetas y archivos:

- Carpeta */source*, que contiene el código fuente de los proyectos Java:
 - *dependencias*, proyecto con las dependencias del editor.
 - *editorCss*, proyecto con el editor de estilos.
 - *es.urjc.editorDocumentos*, proyecto con el código del modelo de datos.
 - *es.urjc.editorDocumentos.diagram*, proyecto con el editor de documentos.
 - *es.urjc.editorDocumentos.diagram.paso*, proyecto con el editor de los componentes del paso.
 - *es.urjc.editorDocumentos.edit*, proyecto con el código de serialización del modelo de datos.
 - *gestionDocumentos*, proyecto con el repositorio Web.
 - *libreriaGenerador*, proyecto la librería de generación de documentos.
 - *modelodatos.emf.estructuras*, proyecto con el modelo de datos de las estructuras HTML.
 - *modelodatos.emf.estructuras.edit*, proyecto con el código de serialización de las estructuras HTML.
 - *modelodatos.emf.estructuras.editor*, proyecto con el código del editor de estructuras HTML.
 - *Servers*, proyecto con la configuración del servidor Tomcat.

- Carpeta **/execute**, que contiene:
 - *editorDocumentos*, carpeta con el editor.
 - *gestionDocumentos*, carpeta con un fichero *.war* del repositorio.
- Carpeta **/examples**, que contiene ejemplos de documentos creados por el editor.
- Carpeta **/db**, que contiene un script de la configuración de la base de datos.
- Carpeta **/memory**, que contiene la memoria del proyecto en formato *.pdf*.
- Y el archivo **Readme.txt**, que vuelve a describir el contenido del CD-ROM.

MANUAL DE USUARIO: Editor

El siguiente manual de usuario tiene como misión identificar las operaciones que puede realizar el profesor sobre la aplicación de escritorio.

Primera pantalla:

- Ir a menú: **File**, **New** y **Documento Diagram**. Apareciendo un diálogo *Create Documento Diagram*, donde poder modificar el nombre del diagrama, **File:** y su localización en nuestro PC. Después, pulsar **Next**, *Create Documento Domain Model*, donde poder modificar el nombre del documento, **File:** y su localización. Pulsar **Finish**.

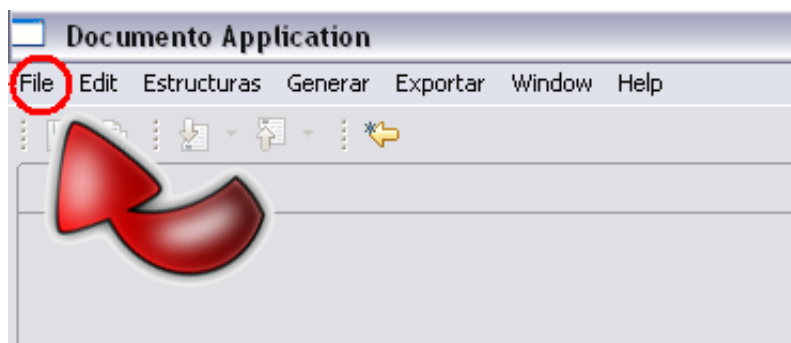
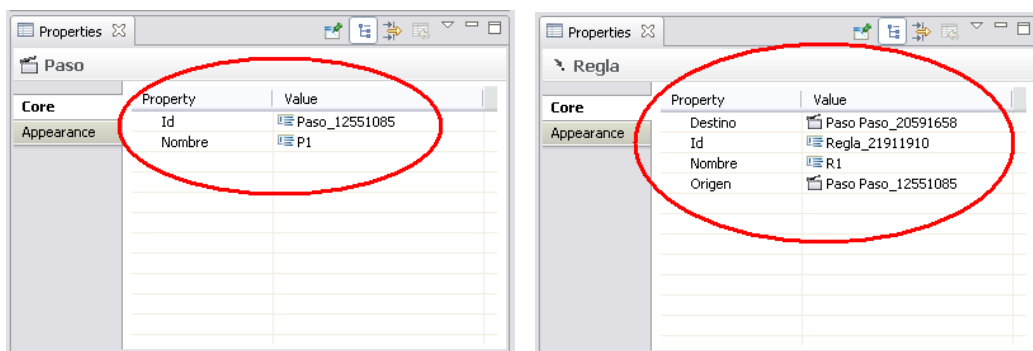


Figura A.1: Primera pantalla de la aplicación.

- **Paso.** Hacer click en **Paso**, y hacer click, de nuevo, en el diagrama de navegación. Se añade el paso, y deja el cursor para poner el nombre del paso, en la cabecera de este.
Se pueden editar sus propiedades, seleccionando el paso y, en la vista *Properties*, en **Core**, modificar *Id* y *Nombre*.
- **Regla.** Para añadir una regla, previamente se han debido incluir al menos dos pasos. Hacer click en **Regla**, y hacer click, de nuevo, en el diagrama de navegación, uniendo un paso origen y otro destino. Deja, también, el cursor para poner el nombre de la regla.
Se pueden editar sus características, seleccionando la regla y, en la vista, *Properties*, en **Core**, modificar *Destino*, *Id*, *Nombre* y *Origen*.



(a) Paso

(b) Regla

Figura A.4: Diálogo para editar propiedades de los pasos y reglas de navegación.

Se puede añadir el estado de conocimiento, seleccionando dicha regla mostrando el menú contextual (botón derecho), y pulsando **EstadoDeConocimiento**. Aparece un nuevo diálogo, *Estado de Conocimiento*, donde se podrá pulsar **Añadir** y **Borrar**, para crear o eliminar los estados de conocimiento. Por último, pulsar **Aceptar** para guardar cambios, o **Cancelar** para no hacerlo.

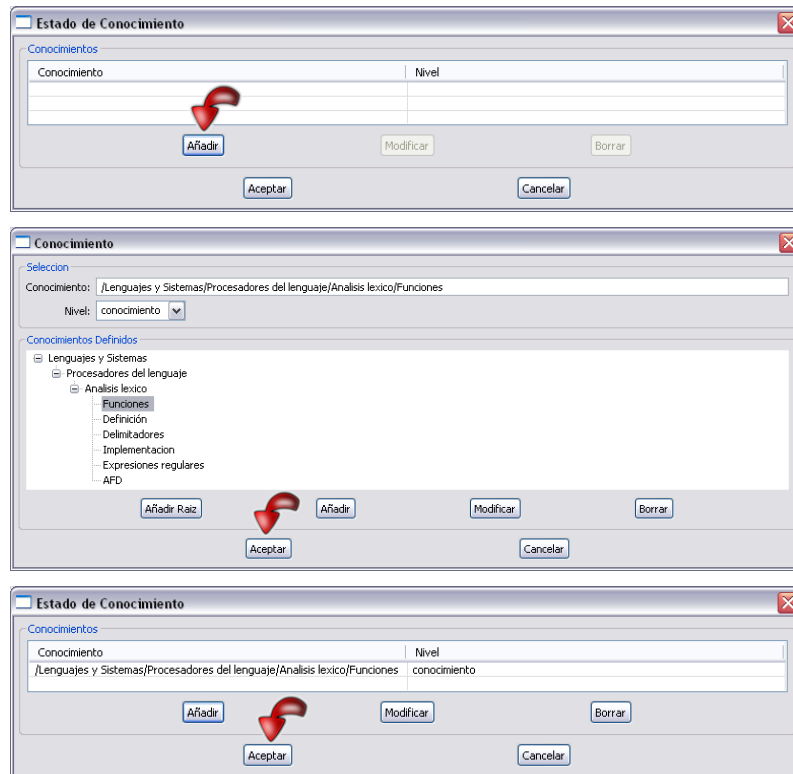


Figura A.5: Diálogo para editar el estado de conocimiento de la regla

Edición de pasos

En la *Caja de herramientas*:

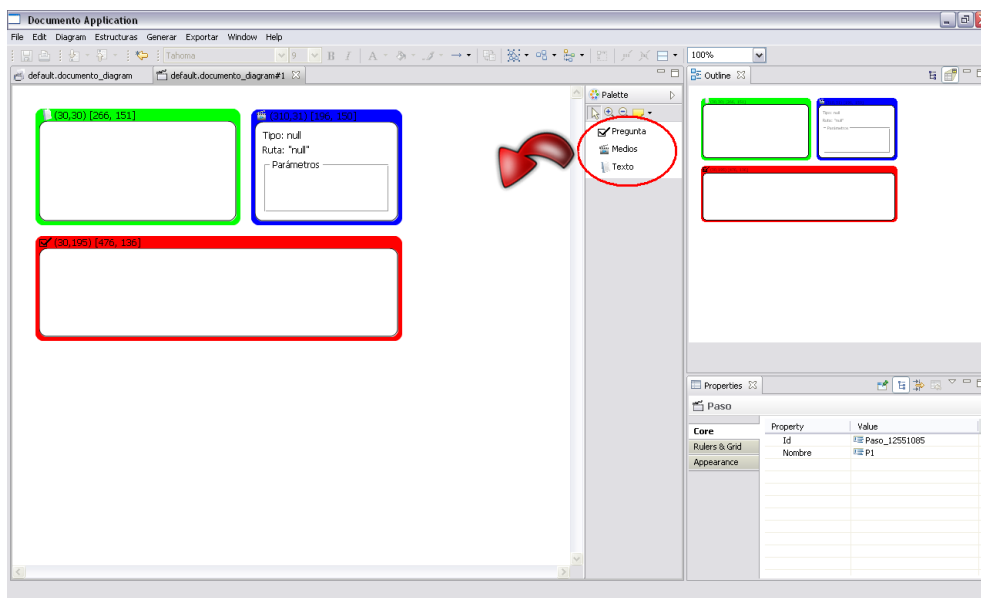


Figura A.6: Diálogo para añadir componentes: narrativo, multimedia y evaluador.

- **Pregunta.** Hacer click en **Pregunta**, y hacer click, de nuevo, en el diagrama de edición del paso. Se añade el componente evaluador.

Se pueden editar sus propiedades, seleccionando el componente y, en la vista *Properties*, en **Core**, modificar *Alto*, *Ancho*, *Clase CSS*, *Pos X* y *Pos Y*.

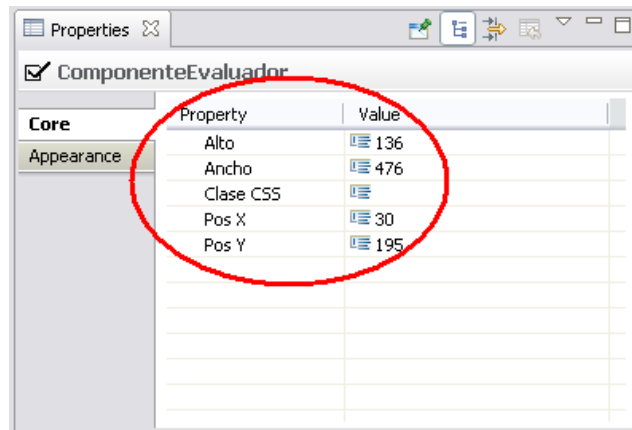


Figura A.7: Diálogo para editar las *propiedades* del componente evaluador.

Se puede añadir el estado de conocimiento, seleccionando dicho componente mostrando el menú contextual (botón derecho), y pulsando **EstadoDeConocimiento**. Aparece un nuevo diálogo, *Estado de Conocimiento*, donde se podrá pulsar **Añadir** y **Borrar**, para crear o eliminar los estados de conocimiento. Por último, pulsar **Aceptar** para guardar cambios, o **Cancelar** para no hacerlo.

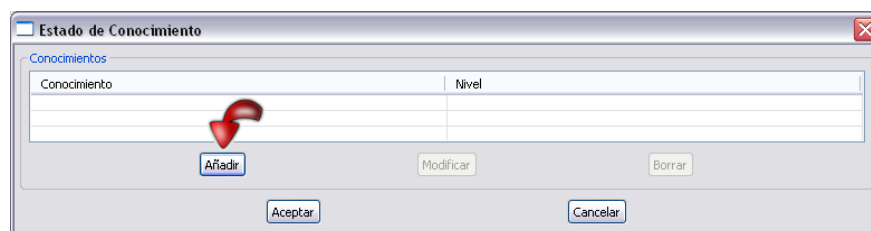


Figura A.8: Diálogo para editar el estado de conocimiento del componente evaluador.

También, se pueden modificar sus características, seleccionando dicho componente mostrando el menú contextual (botón derecho), y pulsando **Modificar**. Aparece un nuevo diálogo, *Componente Evaluador*:



Figura A.9: Menú contextual: Modificar.

Se podrá:

- En *Clase CSS*, escribir la clase del documento .css, del componente.
- En *Enunciado*, escribir un enunciado, con **Añadir Parrafo**, **Modificar Parrafo** y **Borrar Parrafo**. Si se pulsa añadir o modificar, aparece un nuevo diálogo, *Parrafo*, donde añadir los párrafos.
- En *Tipo*, elegir tipo de pregunta: Test, Un Valor o Rellenar Huecos.
 - **Test**, escribir *Respuesta Test*, con **Añadir**. Aparece nuevo diálogo, *Respuesta Test*, donde:
 - ◊ En *Clase CSS*, escribir clase del documento .css, de la respuesta.
 - ◊ En *Respuesta Correcta*, elegir si es o no la respuesta correcta.
 - ◊ Los botones: *Añadir Parrafo*, *Modificar Parrafo* y *Borrar Parrafo*, para escribir las repuestas. Añadir o modificar muestran de nuevo el diálogo, *Parrafo*, donde añadir los párrafos.
 - ◊ En *FeedBack*, escribir el feedback de las respuestas.
 - **Un Valor**, escribir la *Respuesta Valor Correcto* y su *Feedback*.

- **Rellenar Huecos**, escribir *Respuesta de Rellenar Huecos* con los botones **Añadir**, **Modificar** y **Borrar**. Donde, al pulsar en añadir o modificar, aparece nuevo diálogo, *Respuesta de Rellenar Huecos*, donde:
 - ◊ En *Clase CSS*, escribir clase del documento .css, de la respuesta.
 - ◊ En *Texto*, con los botones **Añadir** y **Modificar** donde, al pulsar en , aparece otro diálogo, *Texto con Css*, escribir el *Texto* de las respuestas y elegir su tipo *Css*. Además, de los botones **Borrar** que borra el texto añadido y seleccionado, y el botón **Valor**, que convierte dicho texto en un hueco (valor que el alumno tendrá que averiguar).
 - ◊ En *Resultado*, mostrar el párrafo resultante de añadir textos y valores.
 - ◊ En *Feedback*, escribir el feedback de las respuestas.
- Pulsar **Aceptar** para guardar cambios, o **Cancelar** para no hacerlo.

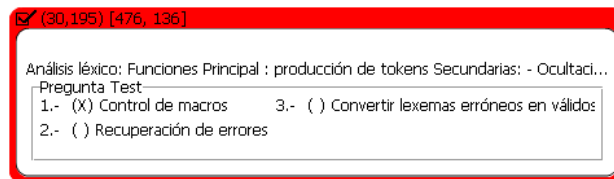


Figura A.10: Resultado de la edición del componente evaluador.

- **Medios**. Hacer click en **Medios**, y hacer click, de nuevo, en el diagrama de edición del paso. Se añade el componente multimedia.
Se pueden editar sus propiedades, seleccionando el componente y, en la vista *Properties*, en **Core**, modificar *Alto*, *Ancho*, *Clase CSS*, *Path Absoluto*, *Pos X*, *Pos Y* y *Tipo*.

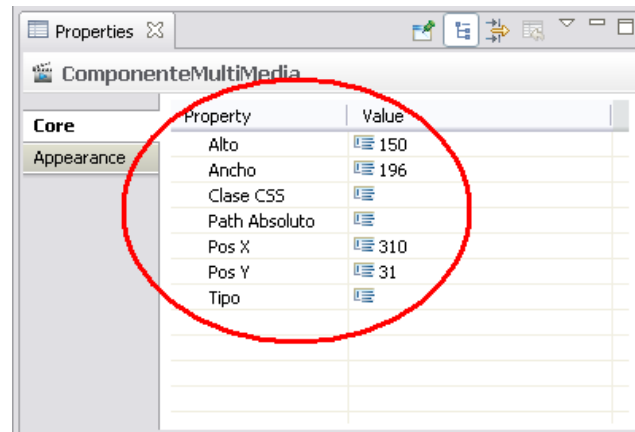


Figura A.11: Diálogo para editar las *propiedades* del componente multimedia.

También, se pueden modificar sus características, seleccionando dicho componente mostrando el menú contextual (botón derecho), y pulsando **Modificar**. Aparece un nuevo diálogo, *Componente Multimedia*:

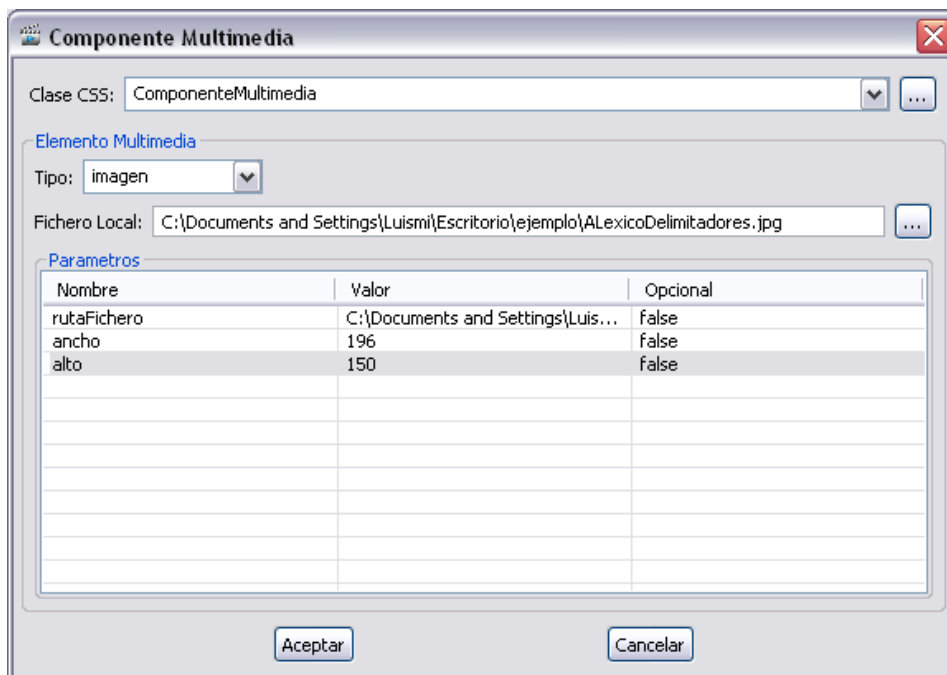


Figura A.12: Menú contextual: Modificar.

Se podrá:

- En *Clase CSS*, escribir la clase del documento .css, del componente.

- En *Elemento multimedia*, elegir *Tipo* (imagen, sonido, video, videoRealPlayer, videoYoutube, videoFlash y videoQuicktime), añadir *Fichero Local* y modificar los valores de los *Parámetros* que se muestran.
- Pulsar **Aceptar** para guardar cambios, o **Cancelar** para no hacerlo.

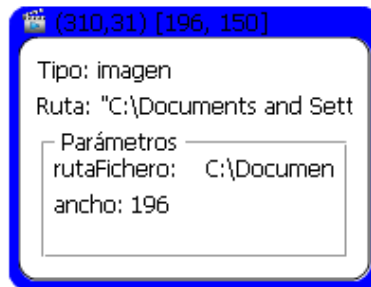


Figura A.13: Resultado de la edición del componente multimedia.

- **Texto.** Hacer click en **Texto**, y hacer click, de nuevo, en el diagrama de edición del paso. Se añade el componente narrativo.

Se pueden editar sus propiedades, seleccionando el componente y, en la vista *Properties*, en **Core**, modificar *Alto*, *Ancho*, *Clase CSS*, *Pos X* y *Pos Y*.

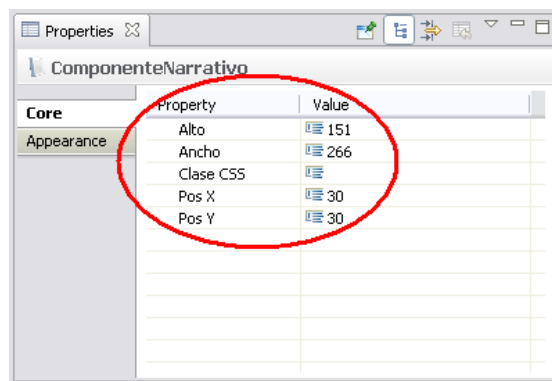


Figura A.14: Diálogo para editar las *propiedades* del componente narrativo.

También, se pueden modificar sus características, seleccionando dicho componente mostrando el menú contextual (botón derecho), y pulsando **Modificar**. Aparece un nuevo diálogo, *Componente Narrativo*:

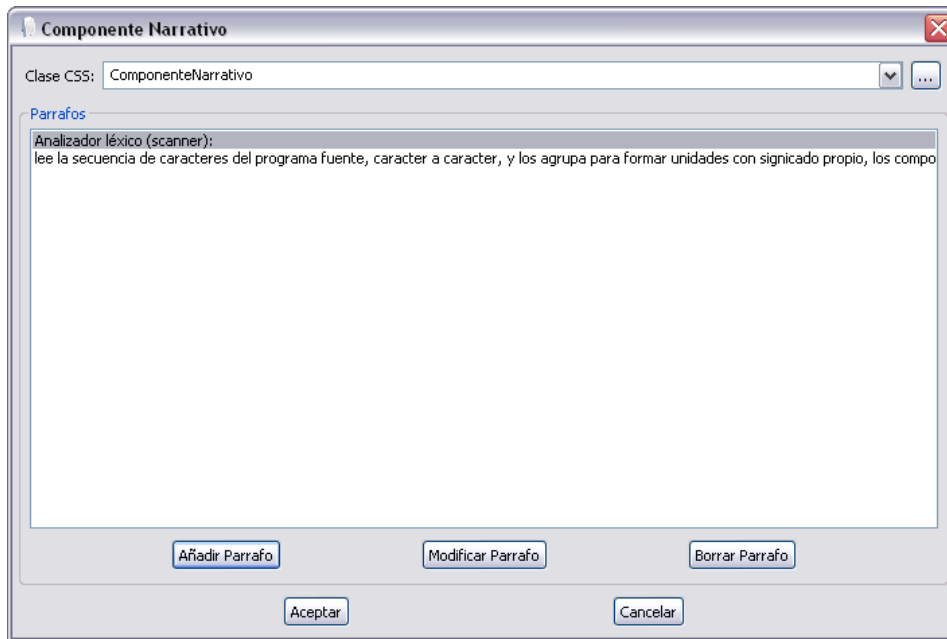


Figura A.15: Menú contextual: Modificar.

Se podrá:

- En *Clase CSS*, escribir la clase del documento .css, del componente.
- En *Párrafos*, pulsar **Añadir Párrafo**, **Modificar Párrafo** y **Borrar Párrafo**, para escribir el texto del componente narrativo. Pulsar **Aceptar** para guardar cambios, o **Cancelar** para no hacerlo.

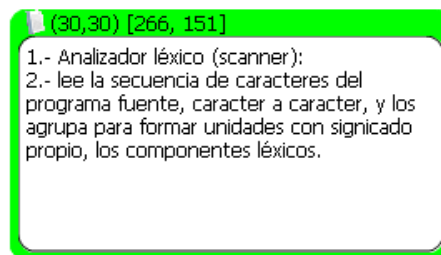


Figura A.16: Resultado de la edición del componente narrativo.

Exportación Web

Una vez contruidos todos los pasos y su contenido, y navegación, se puede exportar a la Web. Para ello se creará un documento comprimido, que contenga toda la información necesaria para visualizarlo.



Figura A.17: Diálogo para exportar a la Web.

Generación de documentos de visualización Web

También, una vez contruidos todos los pasos y su contenido, y navegación, se puede generar un documento, para su visualización en la Web.



Figura A.18: Diálogo para generar un documento offline.

Edición del documento de estructuras

Desde cualquiera de las dos vistas, se puede editar el documento de estructuras, para los componentes multimedia, desde el menú *Estructuras*, con submenú también *Estructuras*.

MANUAL DE USUARIO: Repositorio

El siguiente manual de usuario tiene como misión identificar las operaciones que puede realizar el usuario (docente, alumno y administrador), sobre la aplicación web.

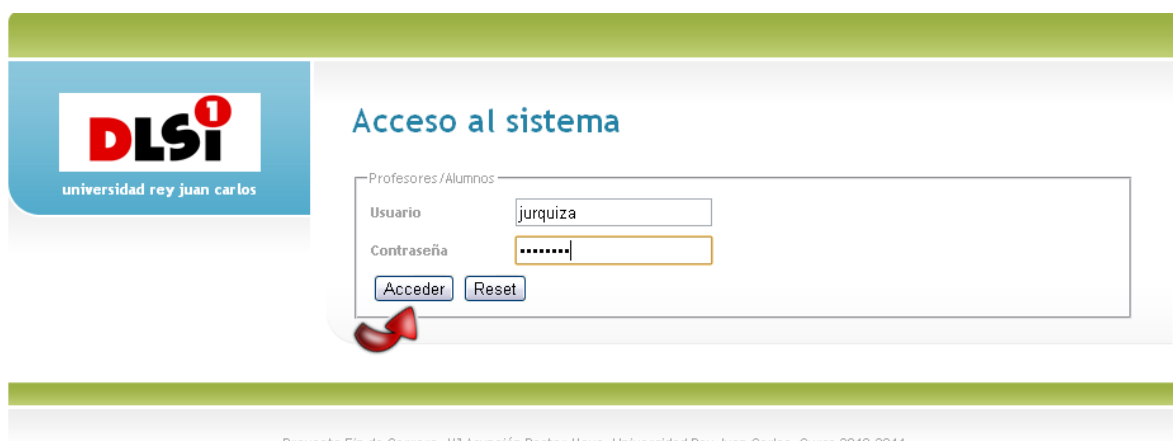


Figura B.1: Pantalla de acceso a la aplicación Web

Docente

Las secciones de la aplicación web a las que puede acceder son:

- Gest. Documentos:
 - **Seleccionar archivo.** Se selecciona un archivo comprimido .zip.

- **Subir documento.** Se pulsa y la aplicación web sube dicho documento al servidor.



Figura B.2: Pantalla: Gestión Documentos

- **Eliminar documentos.** Se selecciona un documento de la tabla y se pulsa un aspa roja. El sistema pedirá confirmación y “Aceptar”/“Cancelar”.



Figura B.3: Eliminar documento

- **Generar documentos offline secuenciales/adaptativos.** El sistema lanzará unos procesos, devolviendo una respuesta satisfactoria o no.

- **Generar documentos online secuenciales/adaptativos.** Al igual que en la generación offline, se lanzarán los procesos de generación, en este caso online, y el sistema devolverá una respuesta satisfactoria o no.



Figura B.4: Generar documento offline u online

■ Gest. Grupos:



Figura B.5: Pantalla: Gestión Grupos

- **Crear grupo.** Se introduce el nombre del grupo y Crear grupo.
- **Eliminar grupo.** Se selecciona un grupo de la tabla y se pulsa un aspa roja. El sistema pedirá confirmación y “Aceptar”/“Cancelar”.
- **Editar grupo.** Se selecciona un grupo de la tabla y se pulsa el botón junto al aspa roja (lápiz sobre papel). El sistema permite la modificación del nombre y “Modificar”.

■ Asoc. Grupos y Documentos.

- **Asociar.** Se selecciona un documento y uno o varios grupos y “Asociar”.



Figura B.6: Pantalla: Asoc. Grupos y Documentos

- Asoc. Grupos y Alumnos.
 - **Asociar.** Se selecciona un grupo y uno o varios alumnos y “Asociar”.



Figura B.7: Pantalla: Asoc. Grupos y Alumnos

Alumno

- Ver documentos.



Figura B.8: Pantalla: Ver documentos

- **Visualizar documento offline.** Se selecciona el documento que se quiere visualizar y pulsar en el documento de la columna *Path offline*. Se descarga en una ruta del PC del alumno, entonces, se tendrá que descomprimir y, una vez hecho esto, ejecutar con un navegador web el documento *index.html*.

- **Visualizar documento online.** Se selecciona el documento que se quiere visualizar y pulsar en el documento de la columna *Path online*. Se abrirá un nuevo navegador web, que mostrará el primer paso del documento.

Administrador

- Gestión usuarios.

Gest. Usuarios desconexión

DLSI
universidad rey juan carlos

Gestión de Usuarios

Usuario (nick)*

Correo electrónico*

Contraseña*

Repita contraseña*

Elija perfil* Profesor ▾

Los campos marcados con asterisco (*) son obligatorios.

Usuarios del sistema

Usuario (nick)	Correo electrónico	Perfil		
luismi	luismi@gmail.com	alumno		
asun	asuncion@urjc.es	profesor		
ma.pastor	ma.pastor@alumnos.urjc.es	alumno		
Jurquiza	Jurquiza@urjc.es	profesor		

Proyecto Fin de Carrera, M^a Asunción Pastor Moya. Universidad Rey Juan Carlos. Curso 2010-2011.

Figura B.9: Pantalla: Gestión Usuarios

- **Dar de alta.** Se rellena el formulario de pantalla y “Dar de alta”.
- **Eliminar usuario.** Se selecciona un usuario y se pulsa el aspa roja. El sistema pedirá confirmación y “Aceptar”/“Cancelar”.
- **Modificar.** Se selecciona un usuario, el sistema permite modificar ciertas características y “Modificar”.

LENGUAJES DE MARCADO

Los lenguajes de marcado surgen a partir del nacimiento de SGML (Standard Generalized Markup Lenguaje) que es una norma de estandarización derivada de GML (Generalized Markup Lenguaje), la cual define que el etiquetado debe describir la estructura del documento y otros atributos. Es decir, el lenguaje de marcado es un conjunto de reglas que definen las partes que conforman un documento, pero que no pertenecen al contenido en sí. Por ejemplo, un documento puede contener diferentes elementos como son capítulos, títulos, párrafos, pies de página, etc. y son estos elementos los que se definen por el lenguaje de marcado. Los lenguajes de marcado se sustentan en el uso de etiquetas, una etiqueta indica el principio de un elemento y otra el final del mismo.

El lenguaje de marcado cumple con dos objetivos esenciales para diseñar y procesar un documento digital:

1. Separa un texto en los elementos en los que se compone, como por ejemplo un párrafo, un capítulo, etc.
2. Especifica las operaciones tipográficas y funciones que debe ejecutar el programa visualizador sobre dichos elementos. Las operaciones tipográficas son instrucciones de formato que se aplican a cada uno de los elementos de un documento digital, por ejemplo, imprimir un título en itálicas.

En función del objetivo que pretenda cumplir un lenguaje de marcado pueden identificarse lenguajes procedurales o lenguajes estructurales.

Lenguajes procedurales. Las anotaciones de los lenguajes procedurales describen la forma y el significado de las operaciones tipográficas a ser aplicadas a cada uno de los elementos del documento. Por ejemplo, una regla del lenguaje procedural indicaría que el título de la sección de un texto debe ser impreso en una sola línea con una fuente de seis puntos más grande que el resto del texto, con objeto de que los lectores puedan inferir que es el título.

Lenguajes Estructurales. En los lenguajes estructurales las anotaciones únicamente describen la estructura lógica de un documento digital, no su tipografía.

XML se ajusta perfectamente a las necesidades de un formato de intercambio, y concretamente XML Schema aumenta las posibilidades de este metalenguaje. XSLT se hace relativamente pesado a la hora de aprenderlo, y al contrario, XML Schema resulta de muy fácil utilización. Esto hace pensar en que XSLT debe evolucionar hacia una mayor usabilidad. JDOM y SAX resultan vitales a la hora de trabajar con XML en el mundo Java.

Java presenta una facilidad en su aprendizaje y en su uso, que pocos lenguajes proporcionan. Permite crear interfaces gráficas de usuario muy fácilmente. Además el hecho de que la portabilidad y distribución de aplicaciones no presente un problema, le ha hecho llegar a ser uno de los lenguajes más demandados en la actualidad. Un punto más en su favor es el hecho de que posee una colección de API's de todo tipo, que facilitan la resolución de problemas simples, que de otra forma se convertirían en escollos que solventar.

Metalinguajes (*XML*)

La Real Academia Española define “Metalenguaje” como: “Lenguaje utilizado para describir un sistema de lenguaje de programación”.

XML, que son las siglas del inglés eXtensible Markup Language (lenguaje de marcado ampliable o extensible) desarrollado por el World Wide Web Consortium (W3C), se ajusta perfectamente a la definición de metalenguaje, ya que es un lenguaje hecho para poder construir otros lenguajes con él. Es una versión simple de SGML y se encuadra dentro de los lenguajes estructurales.

Actualmente XML se ha convertido en el formato preferido para almacenar datos semiestructurados. Los ficheros de texto que utilicen XML para su construcción poseen dos estructuras, una lógica y otra física. Un documento XML es “Bien formado” si cumple las especificaciones sintácticas del estándar XML, es decir, tiene una estructura física correcta. Un documento XML es “Válido” si además de ser “Bien formado” sigue una estructura y una semántica especificada por un DTD o XML Schema, es decir, posee una estructura lógica que encaja en la definida por la DTD o el XML Schema.

Esquemas (*XML Schema y DTD's*)

Un Esquema es un documento de definición estructural. Permite concretar que vocabulario será utilizado y la semántica del documento SGML. En un principio se utilizaron las DTD's para especificar restricciones en la estructura. El DTD puede ser incluido dentro del propio documento o en un fichero externo.

Los XML Schema surgieron después de las DTD's y como sustitutos de ellas. Un XML Schema suple los puntos débiles de las DTD's y posee mejores y mayores capacidades para la definición de estructuras de documentos XML, como la posibilidad de declarar tipos de datos.

Glosario

A

Apache Struts es una herramienta de soporte para el desarrollo de aplicaciones web bajo el patrón MVC bajo la plataforma Java EE (Java Enterprise Edition).

API [Application Programming Interface] es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Archivo Excel es un archivo escrito con la aplicación *Microsoft Excel*, para manejar hojas de cálculo, desarrollada y distribuida por Microsoft.

Articulate compañía que vende diferentes productos e-Learning.

AWT [Abstract Window Toolkit] es un kit de herramientas de gráficos, interfaz de usuario, y sistema de ventanas independiente de la plataforma original de Java.

B

BD [Base de Datos] es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

bytecode lenguaje máquina de la maquina virtual Java.

C

Claroline es una plataforma de aprendizaje y trabajo virtual (eLearning y eWorking) de código abierto y software libre.

CourseLab es una herramienta gratuita para construir contenidos educativos y cursos.

D

DAO [Data Access Object] es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo.

Dokeos es una plataforma de aprendizaje.

DOM [Document Object Model] es un API que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos.

DWR [Direct Web Remoting] es una librería de código abierto escrita en Java que permite escribir aplicaciones web con Ajax.

E

e-Learning se puede definir como el uso de las tecnologías multimedia para desarrollar y mejorar nuevas estrategias de aprendizaje.

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar aplicaciones.

EdiLim es una herramienta que permite la elaboración de materiales educativos.

Efront es un completo sistema de e-learning con una interface agradable.

Exelearning es una herramienta de código abierto que permite generar contenidos educativos.

F

Feedback es un proceso por el que una cierta proporción de la señal de salida de un sistema se redirige de nuevo a la entrada.

Freemind es una herramienta que permite la elaboración de mapas mentales o conceptuales.

G

Grafo es un conjunto de objetos llamados vértices o nodos unidos por enlaces llamados aristas o arcos, que permiten representar relaciones binarias entre elementos de un conjunto.

GUI [Graphical User Interface] es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

H

HotPotatoes es una herramienta que te permite elaborar ejercicios interactivos basados en páginas Web.

I

Ilias es una herramienta web para la educación virtual.

Interfaz declaración de comportamientos necesarios de implementar en la clase que la utilice.

J

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90.

JClíc es una herramienta para la creación de actividades educativas.

JSP es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

M

Maquina virtual de Java programa en lenguaje nativo del sistema que traduce instrucciones desde el lenguaje Java compilado al lenguaje nativo del sistema.

Moddle es el sistema de educación de código abierto más popular.

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Método de enseñanza es el conjunto de momentos y técnicas lógicamente coordinados para dirigir el aprendizaje del alumno hacia determinados objetivos.

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario.

O

offline sin conexión a Internet.

online con conexión a Internet.

P

PC es una microcomputadora, de tamaño medio, diseñada en principio para ser usada por una sola persona a la vez (aunque hay sistemas operativos que permiten varios usuarios simultáneamente, lo que es conocido como multiusuario).

Plug-in es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API.

POJO [Plain Old Java Object] es una sigla creada por Martin Fowler, Rebecca Parsons y Josh MacKenzie en septiembre de 2000 y utilizada por programadores Java para enfatizar el uso de clases simples y que no dependen de un framework en especial.

S

Sigmer Technologies compañía que se dedica a crear herramientas de e-Learning personalizadas.

T

Técnica de enseñanza manera de utilizar los recursos didácticos para un efectivo aprendizaje educando.

Tiles es un framework de plantillas para simplificar el desarrollo de aplicaciones web, interfaz de usuario.

Toolbook es una herramienta de autor que permite realizar todo tipo de contenidos, si bien su principal objetivo es la creación de CBTs (Computer Based Training o formación basada en ordenador) y algunos kioscos (aplicaciones jerárquicas que conducen al usuario de forma interactiva).

U

UML [Unified Modeling Language] es el lenguaje de modelado de sistemas software más conocido y utilizado en la actualidad.

X

XMI [XMI o XML Metadata Interchange] es una especificación para el Intercambio de Diagramas.

Xml [eXtensible Markup Language] es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

Bibliografía

- [1] AENOR. Accesibilidad en aplicaciones de escritorio. <http://www.udc.es/fcs/es/web-to/terapia/asignaturas/toyafam/08tema/UNE139802-2003.pdf>, Junio 2011.
- [2] Mariana Affronti. Una modalidad de enseñanza distinta: e-learning. <http://portal.educ.ar/debates/eid/informatica/testimonios/una-modalidad-de-ensenanza-dis.php>, Junio 2011.
- [3] Pablo Bongiovanni. educomunicacion.com. <http://www.educomunicacion.com/2009/06/aplicaciones-de-codigo-abierto-para.html>, Junio 2011.
- [4] Julio Cabero. Bases pedagógicas del e-learning. <http://www.uoc.edu/rusc/3/1/dt/esp/cabero.pdf>, Junio 2011.
- [5] Universidad Rey Juan Carlos. Campus virtual urjc. <http://cv.urjc.es/campusvirtual/recursosparaprofesores/recursoswindowsprofesores5.php>, Junio 2011.
- [6] Universidad Rey Juan Carlos. Campus virtual urjc. <http://cv.urjc.es/campusvirtual/sobree-learning/index.php>, Junio 2011.
- [7] Jesús de Ceintec. El e-learning como herramienta de desarrollo. <http://www.elearningworkshops.com/modules.php?name=News&file=article&sid=524>, Junio 2011.
- [8] Eclipse Foundation. Eclipse modeling framework project (emf). <http://www.eclipse.org/emf/>, Junio 2011.

-
- [9] Eclipse Foundation. Eclipse, web oficial. <http://www.eclipse.org/>, Junio 2011.
- [10] Eclipse Foundation. Gef (graphical editing framework). <http://www.eclipse.org/gef/>, Junio 2011.
- [11] Eclipse Foundation. Graphical modeling project (gmp). <http://www.eclipse.org/modeling/gmp/>, Junio 2011.
- [12] Eclipse Foundation. Jface - eclipsepedia. <http://wiki.eclipse.org/JFace>, Junio 2011.
- [13] Eclipse Foundation. Swt: The standard widget toolkit. <http://www.eclipse.org/swt/>, Junio 2011.
- [14] The Apache Software Foundation. Apache struts. <http://struts.apache.org/>, Junio 2011.
- [15] The Apache Software Foundation. Apache tomcat. <http://tomcat.apache.org/>, Junio 2011.
- [16] The Apache Software Foundation. Struts tiles. <http://struts.apache.org/1.x/struts-tiles/index.html>, Junio 2011.
- [17] Eric J. Fraude. *Ingeniería del Software. Una Perspectiva orientada a objetos*. RA-MA D.L, 2003.
- [18] Ángel Retamar. Ajax en aplicaciones j2ee con dwr. http://www.javahispano.org/contenidos/es/ajax_en_aplicaciones_j2ee_con_dwr/, Junio 2011.
- [19] Oracle. Java se technologies - database. <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136101.html>, Junio 2011.
- [20] Oracle. Javaseer pages technology. <http://www.oracle.com/technetwork/java/javaee/jsp/index.html>, Junio 2011.
- [21] Richard Bromley Patrick D.T. O'Connor with David Newton. *Practical reliability engineering*. Chischester: John Wiley & Sons, 2004.

-
- [22] Juan Fernández Rodríguez. Dwr: Easy ajax for java. <http://www.docstoc.com/docs/23944790/DWR-Easy-Ajax-for-Java>, Junio 2011.
- [23] World Wide Web Consortium (W3C). World wide web consortium. <http://www.w3c.es/>, Junio 2011.
- [24] w3schools.com. Html <iframe>tag. http://www.w3schools.com/tags/tag_iframe.asp, Junio 2011.
- [25] w3schools.com. Javascript tutorial. <http://www.w3schools.com/js/default.asp>, Junio 2011.
- [26] w3schools.com. Sql tutorial. <http://www.w3schools.com/sql/default.asp>, Junio 2011.
- [27] w3schools.com. w3schools.com: Xml tutorial. <http://www.w3schools.com/xml/default.asp>, Junio 2011.
- [28] w3schools.com. Xml dom tutorial. <http://www.w3schools.com/dom/default.asp>, Junio 2011.
- [29] Webyog. Sqlyog. <http://www.webyog.com/en/>, Junio 2011.
- [30] María Rosa Santiso Fernández y Begoña González González. Diseño multimedia en e-learning para el ámbito universitario. http://www.nosolousabilidad.com/articulos/multimedia_elearning.htm, Junio 2011.
- [31] Andrew Hunt y David Thomas. *The Pragmatic Programmer*. Addison-Wesley Professional, 1999.
- [32] NAGI SIOUFI: Presidente y director general de Auralog. Los efectos online en la formación de los trabajadores. <http://www.elearningworkshops.com/modules.php?name=News&file=article&sid=499>, Junio 2011.
- [33] MySQL y Oracle. Mysql. <http://www.mysql.com/>, Junio 2011.
- [34] Sun y Oracle. Java.com: Java y tú. <http://java.com/es/>, Junio 2011.