

**Universidad
Rey Juan Carlos**

ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

Curso Académico 2010/2011

Proyecto de Fin de Carrera

**Juego adaptativo para la ayuda a los
desplazamientos en transporte público para
usuarios con necesidades especiales**

Autor: Silvia Calvo González

Tutor: Estefanía Martín Barroso

PROYECTO FIN DE CARRERA EN SOPORTE CD

RESUMEN

Durante nuestra vida realizamos gran variedad de actividades que repetimos constantemente asociadas a diferentes contextos. Sin embargo, no todas las personas tenemos las mismas capacidades para realizar dichas actividades. Por ejemplo, mientras que viajar en transporte público, puede resultar trivial para algunas personas puede ser muy duro para otras (p. ej. personas mayores o con limitaciones cognitivas o discapacidades físicas). Esto mismo sucede con muchas otras rutinas diarias.

De forma similar, cada día tenemos que tomar muchas *decisiones*, tanto en situaciones cotidianas (p.e. “¿Qué puedo cocinar hoy para cenar?”) como inesperadas (p.e. “El metro está estropeado. ¿Cómo vuelvo ahora a casa?”). Para algunas personas tomar este tipo de decisiones supone un esfuerzo insignificante, sin embargo, esta tarea puede no ser tan fácil para otras; por ejemplo, suele ser más complicado para personas con necesidades especiales (como las que tienen limitaciones cognitivas).

Por ello a las personas que presentan dificultades para la realización de algunas actividades o para llevar a cabo una toma de decisiones adecuada, es necesario ofrecerles apoyos adicionales en los procesos de aprendizaje. Las nuevas tecnologías, por sus características (repetición, personalización, etc.), son una magnífica herramienta que convenientemente utilizada puede ayudar a lograr que ciertos colectivos logren una mayor independencia en la realización de sus actividades de la vida diaria. Si nos centramos en el caso de las personas que padecen algún tipo de discapacidad intelectual, éstas pueden necesitar más tiempo para realizar ciertas tareas, un mayor número de repeticiones para adquirir algunos conocimientos o ayuda para aprender como tomar decisiones a la hora de resolver situaciones inesperadas, por lo que el uso de *software* adaptado puede ser un apoyo importante en el proceso de aprendizaje o adquisición de capacidades.

En esta línea el presente proyecto ha desarrollado una aplicación adaptativa cuyo objetivo es que usuarios con discapacidades intelectuales puedan utilizar de forma independiente el transporte público. La aplicación se trata de un juego dirigido a usuarios con síndrome de Down que tienen la necesidad de utilizar la red de metro de Madrid diariamente como paso previo a realizar una integración laboral.

El objetivo del juego es facilitar la independencia a la hora de planificar y realizar trayectos, así como que consigan reaccionar y tomar decisiones tanto ante situaciones cotidianas como inesperadas que se pueden producir antes o durante sus viajes en transporte público de un lugar a otro.

Para ello tras realizar un análisis de las características y necesidades concretas de este tipo de usuarios se ha diseñado e implementado una aplicación, a modo de juego, en el que se plantean cuestiones que van aumentando de complejidad según el usuario va obteniendo buenos resultados en el desarrollo del mismo. Las situaciones planteadas a los usuarios en cada momento están adaptadas de acuerdo a sus habilidades personales y necesidades, considerando especialmente sus discapacidades cognitivas y su progreso durante el juego.

*"Dale un pez a un hombre y comerá un día;
enséñalo a pescar y comerá siempre."*

(Proverbio chino)

ÍNDICE GENERAL

1. Introducción.....	I
1.1. Descripción del problema	I
1.2. Estado del arte	II
1.3. Objetivos	VIII
1.4. Metodología de trabajo	IX
1.5. Estructura de la memoria	X
2. Diseño del juego	XI
2.1. Características de los usuarios con síndrome de Down	XI
2.2. Requisitos de usabilidad y accesibilidad	XII
2.3. Aspectos generales del juego	XIV
2.4. Proceso de diseño de la interfaz	XVI
2.5. Estructura y funcionamiento del juego	XX
2.6. Adaptaciones	XXXII
3. Implementación.....	XXXIII
3.1. Estudio de alternativas	XXXIII
3.2. Herramientas y tecnologías empleadas	XXXV
3.3. Organización y estructura de carpetas y ficheros	XXXIX
3.4. Base de datos.....	XLIII
3.5. Implementación de clases	XLIV
3.6. Aplicación para realizar seguimiento de resultados.....	XLVIII
4. Evaluaciones	LI
5. Conclusiones y posibles trabajos futuros	LIII
5.1. Logros alcanzados	LIII
5.2. Posibles trabajos futuros	LV
6. Bibliografía	LVII
7. Apéndices	LIX
7.1. Manual de instalación.....	LIX
7.2. Manual breve para docentes.....	LX
7.3. Cuestionario de evaluación del juego “Metro de Madrid”	LXII

ÍNDICE DE FIGURAS

1.1.	Pantalla del juego “Mis primeros pasos con Pipo”	IV
1.2.	Pantalla del juego “Pizarra digital”	V
1.3.	Pantalla del juego “Precio exacto”	V
1.4.	Pantalla del juego “CITI”	VI
1.5.	Pantallas del juego “Una Imagen Mejor que Mil Palabras”	VII
1.6.	Pantalla del programa “TaskTools”	VII
2.1.	Ejemplos de preguntas de respuesta atómica	XV
2.2.	Ejemplos de preguntas de respuesta compuesta	XV
2.3.	Ejemplos de retroalimentación dada al jugador	XV
2.4.	Diagrama de pantallas del juego presentado a especialistas durante la fase de diseño.	XVI
2.5.	Prototipo final de las pantallas iniciales.....	XVII
2.6.	Primer prototipo de la pantalla de pregunta	XVIII
2.7.	Segundo prototipo de la pantalla de pregunta	XVIII
2.8.	Prototipo final de la pantalla de pregunta	XIX
2.9.	Primer prototipo de la pantalla de respuesta	XIX
2.10.	Segundo prototipo de la pantalla de respuesta	XIX
2.11.	Prototipo final de la pantalla de respuesta	XX
2.12.	Primer prototipo de la pantalla de resumen de puntuaciones	XX
2.13.	Prototipo final de la pantalla de resumen de puntuaciones.....	XX
2.14.	Pantalla de registro de nuevos usuarios	XXI
2.15.	Pantallas de error y confirmación de registro de un nuevo usuario.....	XXI
2.16.	Pantallas de registro de usuario ya existente y de error de registro	XXI
2.17.	Pantallas de bienvenida.....	XXII
2.18.	Estructura de la pantalla de resumen de puntuaciones	XXII
2.19.	Estructura de la pantalla de pregunta	XXIII
2.20.	Pantallas de pregunta del tipo 1 del nivel 4 y su posibilidad de rectificación	XXIV
2.21.	Pregunta tipo 1 del nivel 1	XXV
2.22.	Pregunta tipo 1 del nivel 2	XXV
2.23.	Pregunta tipo 1 del nivel 3	XXV
2.24.	Pregunta tipo 1 del nivel 4	XXV
2.25.	Pregunta tipo 2 del nivel 1	XXVI
2.26.	Pregunta tipo 2 del nivel 2	XXVI
2.27.	Pregunta tipo 2 del nivel 3	XXVI
2.28.	Pregunta tipo 2 del nivel 4	XXVI
2.29.	Pregunta tipo 3 del nivel 1	XXVII

2.30.	Pregunta tipo 3 del nivel 2	XXVII
2.31.	Pregunta tipo 4 del nivel 3	XXVII
2.32.	Pregunta tipo 5 del nivel 1	XXVII
2.33.	Pregunta tipo 5 del nivel 2	XXVIII
2.34.	Pregunta tipo 5 del nivel 3	XXVIII
2.35.	Pregunta tipo 5 del nivel 4	XXVIII
2.36.	Pregunta tipo 6 del nivel 2	XXVIX
2.37.	Pregunta tipo 6 del nivel 4	XXIX
2.38.	Tipos de preguntas generados mediante plantilla	XXX
2.39.	Estructura de la pantalla de respuesta.....	XXXI
2.40.	Pantallas de fin de nivel.....	XXXI
2.41.	Pantalla de resumen de puntuaciones.....	XXXII
3.1.	Patrón modelo-vista-controlador.....	XXXIV
3.2.	Imágenes diseñadas con Microsoft Office Power Point 2007.....	XXXVIII
3.3.	Organización de los principales directorios de la aplicación.....	XXXIX
3.4.	Distintos sistemas reales con estructura de grafo	XLIV
3.5.	Clase <i>GrafMatPeso</i>	XLV
3.6.	Algoritmo de <i>Floyd</i>	XLVI
3.7.	Clase <i>CaminosMinimos</i>	XLVII
3.8.	Ejemplo de pantalla de la aplicación “ <i>Metro de Madrid- Seguimiento de Resultados</i> ”	XLVIII
5.1.	Póster publicado dentro del congreso internacional “ <i>User Modeling, Adaptation and Personalization</i> ” (<i>UMAP 2010</i>).	LIV
7.1.	Pantalla de inicio de ronda	LX
7.2.	Tipo de pregunta de texto libre.....	LXI
7.3.	Tipo de pregunta de selección	LXI
7.4.	Ejemplos de pantallas de retroalimentación a diferentes tipos de preguntas.....	LXI
7.5.	Pantalla con el resumen de puntos obtenidos después de jugar una ronda	LXI
7.6.	Pantalla con el resumen de las puntuaciones obtenidas en todos los niveles	LXII

ÍNDICE DE TABLAS

3.1.	Repartición de tareas entre <i>servlets</i> y <i>JSP</i> en la aplicación	XXXIII
3.2.	Ficheros <i>JSP</i> del directorio <i>web</i>	XIX
3.3.	Ficheros de texto del directorio <i>varios</i>	XL
3.4.	Funciones <i>Javascript</i> implementadas en el fichero <i>utilidades.js</i> del directorio <i>varios</i>	XLI
3.5.	Ficheros <i>JSP</i> del directorio <i>varios</i>	XLII
3.6.	Ficheros <i>servlet</i> del directorio <i>classes</i>	XLII
3.7.	Campos de la tabla “datosjugadores”	XLIII
3.8.	Ficheros <i>JSP</i> del directorio <i>web</i>	XLIX
3.9.	Fichero de texto del directorio <i>varios</i>	XLIX
3.10.	Función <i>Javascript</i> implementada en el fichero <i>utilidades.js</i> del directorio <i>varios</i>	XLIX
3.11.	Fichero <i>JSP</i> del directorio <i>varios</i>	L
3.12.	Fichero <i>servlet</i> del directorio <i>clases</i>	L

1. INTRODUCCIÓN

1.1. DESCRIPCIÓN DEL PROBLEMA

La incorporación al mundo laboral conlleva situaciones y cambios que tienen un alto grado de complejidad e impredecibilidad, convirtiéndose en uno de los pasos más importantes y difíciles en la vida de cualquier persona. Cuando una persona padece una discapacidad intelectual al ya de por sí complejo proceso de integración laboral, se le añaden una serie de dificultades derivadas de la propia discapacidad.

Por ello es necesario que haya un proceso intermedio de transición entre el entorno escolar y el mundo laboral. Con este objetivo son numerosos los centros que ofrecen apoyo para la integración laboral de personas con dificultades de aprendizaje. Los jóvenes llegan a estos centros para preparar su incorporación al mundo laboral, para ello reciben una formación previa para aprender, no un trabajo concreto, sino a ser un trabajador [1]. El apoyo que se les proporciona va dirigido a facilitar la adquisición de las habilidades propias del puesto laboral concreto en el que se va a integrar, así como a entrenarle en otros aspectos complementarios como son la autonomía en los desplazamientos, el control de horarios, las relaciones sociales con los jefes y compañeros de la empresa, y su actividad general diaria. Por ejemplo, para poder ser un trabajador es indispensable, entre otras, disponer de la capacidad de desplazamiento de una manera autónoma y eficiente hasta el centro de trabajo. En grandes ciudades, como Madrid, una forma muy usual de realizar tanto este desplazamiento como otro tipo de viajes por la ciudad (p.e. acudir a estudiar, al cine o a la consulta del médico) es mediante el uso de la red de metro de Madrid.

Los sistemas de transporte público, se encuentran entre los sistemas más complejos encontrados en nuestra sociedad [2]. Para ciertas personas, como las que padecen algún tipo de discapacidad cognitiva o las personas que usan el transporte público en una gran ciudad por primera vez, estos sistemas resultan muy complejos. La libertad para llevar una vida independiente, social o mantener un puesto de trabajo puede estar altamente relacionada con la habilidad para usar estos medios de transporte. Por ejemplo, personas con síndrome de Down, que residen en grandes núcleos urbanos donde el medio de transporte público más común es el metro, a los posibles miedos frente a nuevas experiencias y situaciones al incorporarse a un nuevo puesto de trabajo hay que añadir las dificultades que pueden encontrar en el manejo de la red de metro para acudir a su lugar de trabajo. Deberán tomar decisiones correctamente ante situaciones tanto cotidianas como extraordinarias (p.e. *“Hay una avería en una línea de metro, ¿qué hago para llegar a mi destino?”*, *“Para ir desde la estación “a” hasta la estación “b”, ¿en qué estación tengo que hacer transbordo?”*), prever consecuencias de sus acciones (p.e. *“Si decido continuar mi viaje en la línea “x”, ¿será éste el camino más corto?”*) y entender instrucciones secuenciales (p.e. *“Para ir desde la estación “a” hasta la estación “b” primero debo coger en la estación “a” la línea “x”, al llegar a la estación “c” hacer transbordo a la línea “y” dirección “d” y continuar en esa línea hasta llegar a la estación “b”*”).

Los ordenadores ofrecen una valiosa ayuda a las personas con necesidades especiales. Mediante el uso de las aplicaciones *software* adecuadas se puede realizar un aprendizaje previo para adquirir conocimientos del medio de transporte (p.e. líneas, dónde se puede hacer transbordo) y toma de decisiones tanto ante situaciones cotidianas (p.e. elegir trayecto más corto) como extraordinarias (p.e. avería en el medio de transporte). Así se logra que aprendan y como consecuencia de ese aprendizaje ganen en independencia.

1.2. ESTADO DEL ARTE

Hay multitud de investigaciones y estudios [3] que destacan la utilidad de las nuevas tecnologías para mejorar las habilidades mentales de personas con discapacidad intelectual.

Por ejemplo, se han realizado estudios [4] en los que niños con patologías como autismo, parálisis cerebral, y deficiencias auditivas severas han utilizado programas informáticos para ejercitar conceptos espaciales y corporales involucrados en sus acciones, realizar proyectos de aprendizaje, interactuar con diversos objetos, expresar ideas complejas, y asumir un rol más activo y productivo en su procesos de aprendizaje. También se han aprovechado herramientas informáticas para diseñar estrategias didácticas que buscan estimular el aprendizaje de la lecto-escritura y potenciar habilidades cognitivas en niños con diversos problemas de aprendizaje. En otro estudio, se ha demostrado que mediante el uso de un programa informático por parte de un grupo de discapacitados intelectuales se consiguió que aumentaran sus habilidades de discriminación y, además, transferían esas habilidades a otros contextos. Por último hay un trabajo de la Universidad Complutense de Madrid, en el que exploran las posibilidades del ordenador como instrumento significativo para el entrenamiento de la memoria a través de la enseñanza de estrategias cognitivas de tipo repetitivo. Tomando dos muestras de personas con síndrome de Down (40 en total) obtienen una clara tendencia a la mejora en los sujetos del grupo experimental [3].

Pero, también hay estudios que han tratado de comprobar los beneficios de la utilización del ordenador sin encontrar resultados positivos, demostrando que tanto en las personas con discapacidad como en las que no la tienen, los beneficios no vienen del uso del ordenador por sí mismo (ya que éste es sólo una herramienta), sino del diseño de la aplicación o forma de utilizar el ordenador. Por lo tanto es cuando el ordenador se utiliza como un instrumento cognitivo que estructura, organiza, amplía y desarrolla los procesos mentales del sujeto, cuando su utilización puede ser beneficiosa para los sujetos con discapacidad [3]. Así, mediante la adecuada y adaptada inclusión de recursos tecnológicos se favorece y facilita la comprensión, la experimentación, el análisis, etc., posibilitando por un lado, el entrenamiento de ciertas habilidades relacionadas con la realidad social y la vida cotidiana y, por otro que se encuentren en contacto con el mundo de la tecnología [5].

Por tanto, para lograr una correcta interacción y transmisión del conocimiento es necesario que las adaptaciones tecnológicas se fundamenten en dos aspectos:

- **El hardware:** realizando las adaptaciones necesarias en los dispositivos en entrada y salida (p.e. dispositivos especiales de Braille para usuarios con discapacidades visuales graves).
- **El software:** desarrollando programas para acceder al ordenador, de ayuda a la comunicación (comunicación aumentativa), para el desarrollo del lenguaje o el desarrollo de procesos cognitivos, diseñando interfaces que permitan la adaptación a las características y necesidades de las personas a las que va dirigida la aplicación.

En concreto, en los usuarios con síndrome de Down, las nuevas tecnologías son una herramienta muy útil para facilitar el acceso a la información y el aprendizaje siempre que estos materiales les resulten accesibles. Debido a la personalización y secuenciación de tareas que permite el ordenador el uso de materiales multimedia adaptados puede ayudar a solucionar algunos de los problemas de aprendizaje que presenta este grupo de usuarios [6] [7], por ejemplo:

- Propiciar un mayor interés y actitud positiva hacia el aprendizaje.
- Facilitar la comprensión de los problemas simulados y planteados en contextos de la vida cotidiana. Se deben presentar situaciones de enseñanza que favorezcan la posterior generalización de aprendizajes, simulando, siempre que sea factible, múltiples contextos, lo que proporcionará mejores condiciones para la posterior generalización de los contenidos y habilidades aprendidos. Además, es muy importante que las situaciones se enmarquen dentro de actividades de la vida cotidiana para favorecer la transferencia de conocimiento.
- Amortiguar los problemas de retención al permitir presentar la información de forma sistemática y reiterada.
- Ayudarles a focalizarse en los elementos relevantes de la tarea para mitigar así las dificultades de atención.
- Facilitar el pensamiento abstracto (mediante la inclusión de estructuras dinámicas de presentación: elementos en movimiento, ejemplificaciones, experiencias, etc.).

En la actualidad existen multitud y variadas aplicaciones multimedia que se han desarrollado y/o utilizado como herramienta de apoyo en procesos de aprendizaje o entrenamiento en usuarios con Síndrome de Down.

La gran mayoría de ellas están enfocadas al aprendizaje de conceptos matemáticos, desarrollo de habilidades comunicativas o lecto-escritura, entre otros por lo que están dirigidas a usuarios en edad escolar. Pero, aunque en menor cuantía, también nos encontramos con aplicaciones específicamente encaminadas a favorecer la integración social y laboral de este colectivo.

Para hacer un esbozo general de la situación actual de este tipo de aplicaciones se presentan las características de algunas de ellas, así como los resultados obtenidos en los casos en que se han realizado estudios de su eficacia.

⇒ **“ABC Landia”** [4]

Programa audiovisual e interactivo desarrollado por la Universidad de Antioquia (Colombia). Está enfocado a niños en edades comprendidas entre 4 y 12 años con Síndrome de Down. Su objetivo es estimular el desarrollo de las cuatro habilidades comunicativas básicas: hablar, escuchar, leer y escribir.

Para el diseño se seleccionaron 140 palabras con las que se construyó un inventario de frases y textos, con una estructura gramatical sencilla, de fácil comprensión, y con sentido completo. Se utilizaron textos cortos de carácter narrativo, expositivo e informativo, que describían de manera simple y amena situaciones familiares, sociales, ambientales y culturales.

El objetivo de la aplicación es sumergir al niño en un trabajo simultáneo de lectura y escritura, en el que realiza diversas actividades con una palabra estímulo. El niño trabaja con esta palabra viéndola, escuchándola, pronunciándola, escuchando su propia pronunciación y apreciándola en el contexto de una frase.

Tras realizar un estudio con 20 niños, los principales hitos que se observaron fueron:

- Avances en la capacidad de articulación. La mayoría de los niños aumentaron el número de fonemas articulados correctamente. El interés de los niños en grabar su voz los obliga a elevar el tono habitualmente bajo de sus emisiones, lo cual incide positivamente en la eficacia de su comunicación oral.
- Todos los niños mostraron avances en su proceso de apropiación de la escritura. La utilización del teclado neutraliza las dificultades de coordinación visomotriz implícitas en la escritura manuscrita, lo que puede influir positivamente en la motivación para abordar tareas de escritura, y en la cantidad y calidad de los textos que generan.
- En cuanto a la lectura, destaca el interés por la lectura de imágenes en los más pequeños, y en los niños mayores el seguimiento que hacen a la historia, el reconocimiento de los personajes, y la comprensión que logran de las historias.
- La habilidad de escuchar es quizá la que menos se destacó dado el predominio de la atención visual en estos niños.

⇒ **“Mis primeros pasos con Pipo (Cibal Multimedia)” [7]**

Juana M. Ortega-Tudela y Carlos J. Gómez Ariza, de la Universidad de Jaén, han realizado un trabajo para conocer hasta qué punto el ordenador puede resultar un instrumento útil de enseñanza ayudando a la generalización de conocimientos y habilidades matemáticas a personas con síndrome de Down. En este caso se trata de hacer uso de un juego estándar del que se hace una selección de actividades.

En este estudio participaron 18 niños con síndrome de Down, divididos en dos grupos que presentaban un rendimiento similar en todas las tareas. Cada uno de los grupos fue instruido, con distintas metodologías de enseñanza, en los principios de conteo y cantidad. Un grupo trabajó con material multimedia (“Mis primeros pasos con Pipo”, ver Figura 1.1) y el otro con una metodología tradicional de lápiz y papel donde se imprimieron en color las pantallas del juego. Así, se realizaban los mismos juegos que en ambos grupos pero con la mediación del ordenador en el primero y del profesor en el segundo.

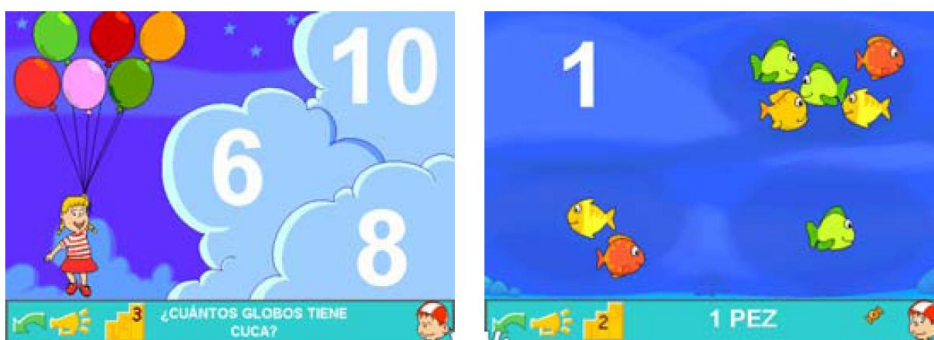


Figura 1.1.- Pantalla del juego “Mis primeros pasos con Pipo”.

Tras el proceso de enseñanza, ambos grupos fueron evaluados mediante una tarea que simulaba una situación de compra. Los resultados sugieren un claro efecto facilitador de la metodología multimedia en la generalización de los conceptos y habilidades básicas de conteo en niños con síndrome de Down.

⇒ **“Pizarra digital”** [5]

Sistema educativo multimedia, diseñado por un equipo multidisciplinar de la Universidad de la Laguna junto con la Asociación Tinerfeña de Trisómicos 21, dentro del contexto de la resolución de los problemas en la enseñanza de las matemáticas y en hacer accesibles las interfaces mediante su adaptación vía *software* y *hardware*.

La pizarra digital (ver Figura 1.2.) está diseñada para facilitar la interacción de niños con problemas de motricidad fina y pretende ser un sistema de ayuda al docente en la detección de errores en la resolución de problemas y algoritmos y su posterior tratamiento, utilizando interfaces accesibles.

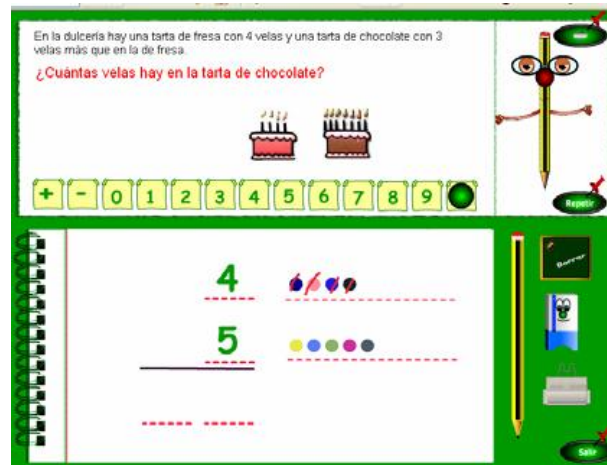


Figura 1.2.- Pantalla del juego “Pizarra digital”.

⇒ **“Precio exacto”** [8]

Desarrollado por el grupo AmILab de la Universidad Autónoma de Madrid, dentro del proyecto HADA (*Hipermedia Adaptativa para la atención a la Diversidad en entornos de inteligencia Ambiental*). Es una aplicación que pretende facilitar el aprendizaje y experimentación del manejo del dinero, presentando para ello situaciones contextualizadas en la vida diaria. La Figura 1.3. presenta un ejemplo de la aplicación.



Figura 1.3.- Pantalla del juego “Precio exacto”.

⇒ **“CITI” (Competencia Intelectual y Tecnología de la Información)** [9]

CITI es un juego promovido por la Fundación Orange y la Fundación Síndrome de Down de Madrid, diseñado para mejorar las capacidades cognitivas de sus usuarios de forma divertida.

Está disponible para PCs y para teléfonos móviles. Presenta distintas actividades que trabajan diferentes habilidades adaptativas de la vida cotidiana: autonomía en el hogar, autocuidado, uso de recursos comunitarios o interés hacia la cultura, sin perder la orientación de juego y diversión (ver Figura 1.4.).

Consta de cuatro actividades de estrategia (ir al cine, a patinar, organizar una fiesta y pasar un día en el museo). En cada una de ellas el jugador debe superar una serie de pruebas para pasar a la siguiente actividad, obteniendo refuerzos si resuelve las actividades. El juego se desarrolla en una ciudad con cinco escenarios (casa, parque, centro comercial, centro cultural y metro) en los que el jugador se mueve contando con la ayuda de un asistente.



Figura 1.4.- Pantalla del juego “CITI”.

⇒ **“Work with Me Training Suite” y “Una Imagen Mejor que Mil Palabras”**[1]

En [1], se presentan dos herramientas. La primera (“*Work with Me Training Suite*”) fue diseñada por Autisme Ltd. Ofrece una gran variedad de lecciones organizadas en cursos, orientadas al empleo, desarrollo del currículum, etc. Está herramienta está orientada a personas con autismo. Se ofrece en soporte CD para ordenador y está estructurado de manera interactiva y visual. La segunda (“*Una Imagen Mejor que Mil Palabras*”) se trata de una herramienta multimedia, en formato DVD, que pretende ofrecer recursos útiles para el trabajo con personas con síndrome de Down y otras discapacidades intelectuales dentro del proceso de aprendizaje y preparación previo a la inserción en el mundo laboral. Es fruto de un proyecto llevado a cabo con la colaboración de la Fundación Projecte Aura de Barcelona y la Asociación de Síndrome de Down de Burgos.

Se presenta como un material educativo sencillo, intuitivo e interactivo. Sus autores la enmarcan dentro de la fase de formación previa al empleo, donde se trabajan las habilidades sociales y autonomía personal (ver Figura 1.5.). Pretende sentar las bases necesarias para poder acceder al mundo laboral, anticipando situaciones que puedan aparecer para poder trabajarlas con anterioridad, favoreciendo que las primeras experiencias de trabajo sean lo más satisfactorias posibles, disminuyendo el riesgo de dificultades, y potenciando la autonomía personal y las habilidades sociales en el contexto laboral.

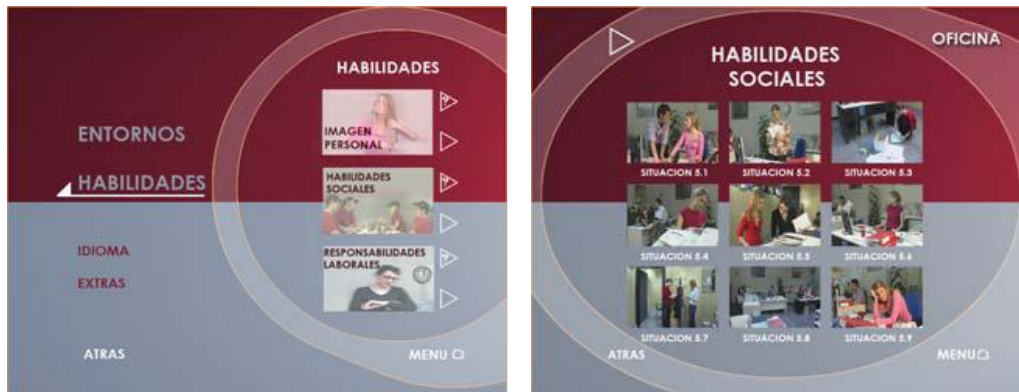


Figura 1.5.- Pantallas del juego “Una Imagen Mejor que Mil Palabras”.

⇒ “TaskTools” [10]

Desarrollado por Gureak Araba S.L., con el objetivo de optimizar un sistema automatizado de ayuda para personas con discapacidad cognoscitiva, específicamente con Síndrome de Down, de ayuda en el ámbito laboral y la vida cotidiana.

Esta aplicación está desarrollada para dispositivos móviles como teléfonos móviles o PDAs. Posee una interfaz multimedia cómoda, flexible, configurable por personal no técnico y adaptada ergonómicamente.

La aplicación contiene una herramienta de generación y gestión de tareas que permite que personas no técnicas de ayuda a este colectivo (tutores personales, psicólogos, familiares) puedan configurarlo de forma fácil y sencilla.

El objetivo principal es que las personas de apoyo a este colectivo configuren de forma cómoda y flexible dicho sistema. Así las personas con síndrome de Down disminuyan su dependencia, aumenten su autonomía en el contexto laboral y mejoren su rendimiento y seguridad en el desempeño de sus tareas de trabajo. La Figura 1.6. muestra distintas pantallas de esta herramienta de gestión de tareas.



Figura 1.6.- Pantalla del programa “TaskTools”.

A modo de conclusión de este estudio decir que hay multitud de iniciativas (principalmente dentro del mundo universitario y de las asociaciones y fundaciones de ayuda a este colectivo) para desarrollar herramientas que sean de utilidad en el entrenamiento de capacidades y habilidades en personas con discapacidad cognitiva, en concreto con síndrome de Down.

La incorporación de juegos multimedia en los procesos de aprendizaje, ha ayudado a evolucionar los recursos con los que cuentan los profesores a la hora de llevar a cabo su trabajo. Son un medio atractivo, por su naturaleza lúdica, que suponen un mecanismo de estímulos para los usuarios, facilitan los procesos de aprendizaje y además pueden ser utilizados, en contextos diferentes a los habituales, dentro del aula. En el caso del *software* desarrollado en forma de juego nos encontramos con aplicaciones que proponen un medio donde el conocimiento se obtiene a través del contenido del juego, y las habilidades cognitivas se desarrollan como resultado de la propia acción de jugar [11].

Si bien es cierto que son minoría las aplicaciones que tienen como objetivo la adquisición de las habilidades necesarias para la posterior incorporación al mundo laboral de este colectivo. Para lo que sin duda es muy útil disponer de herramientas, como la que se va a presentar en este proyecto, que ayudan al entrenamiento de la habilidad de toma de decisiones relacionadas con el transporte.

Puede ser que la ausencia de *software* en esta área sea debido a que la incorporación al mundo laboral de personas con síndrome de Down y otras discapacidades intelectuales es aún minoritaria: un 8,4% se encuentra trabajando frente a un 91,6% que se encuentra sin trabajo [12]. Sin embargo, se están haciendo grandes esfuerzos por parte de asociaciones y fundaciones para que esta situación cambie, ya que hoy en día es incuestionable el hecho de que las personas con Síndrome de Down pueden trabajar, y que este hecho les facilita su incorporación de una forma lo más normalizada posible a la sociedad [13].

1.3. OBJETIVOS

En este proyecto, se presenta un juego adaptativo cuyo objetivo es entrenar a los usuarios con necesidades especiales a usar medios públicos de transporte.

En concreto se ha desarrollado una aplicación multimedia orientada a usuarios con discapacidades cognitivas como el síndrome de Down que están preparando su incorporación al mundo laboral. Se pretende que con el uso del juego conozcan el medio de transporte público por el que se van a mover y que se entrenen en posibles situaciones tanto esperadas como inesperadas que se pueden encontrar al desplazarse de un lugar a otro usando el transporte público, en concreto mediante la red de metro de Madrid.

El objeto principal es desarrollar una herramienta *software* con contenido adaptado, graduado en niveles e interactivo, adecuado para la fase de formación en el manejo de transportes públicos, previa al empleo.

Se pretende proporcionar a los jóvenes con síndrome de Down los recursos necesarios que favorezcan y mejoren las condiciones en las que se enfrentan a sus desplazamientos en metro para así ayudar a la integración laboral, social, incrementar la autonomía personal y mejorar habilidades tales como el autocontrol y la toma de decisiones a través de la adaptación y uso adecuado de las nuevas tecnologías.

Con el uso del juego se pretende que los usuarios adquieran las siguientes habilidades y conocimientos:

- El conocimiento del medio en el que se van a mover, esto es, de la red de metro de Madrid (líneas existentes, identificación del color de cada línea, estaciones que pertenecen a cada línea, estaciones en que se puede hacer transbordo, etc.).
- El aprendizaje para poder seleccionar la mejor ruta entre dos lugares de acuerdo con unas necesidades específicas (p.e. minimizando el número de transbordos o el número de estaciones que se visitan).
- El desarrollo de las capacidades de toma de decisiones y resolución de problemas, mediante el planteamiento a lo largo del juego de situaciones reales y en algunos casos inesperadas que se pueden encontrar en sus desplazamientos (p.e. averías de líneas de metro, cierre de estaciones, etc.). Es prioritario que se produzca una generalización de las habilidades y conocimientos adquiridos con el uso del juego, para que al enfrentarse a ese tipo de situaciones en la vida real tengan los mecanismos suficientes para tomar la decisión más idónea en cada momento.

1.4. METODOLOGÍA DE TRABAJO

Para el desarrollo de este proyecto se ha seguido la siguiente metodología de trabajo:

- Toma de contacto con el mundo de las nuevas tecnologías como herramientas de ayuda a personas con discapacidades intelectuales, profundizando en los juegos multimedia como medio de aprendizaje y entrenamiento en habilidades dirigidos a personas con síndrome de Down.
- Búsqueda bibliográfica de las características y necesidades de los usuarios con síndrome de Down a tener en cuenta en el diseño de juegos accesibles y adaptados.
- Diseño del juego estableciendo la estructura del juego en distintos niveles de dificultad, el tipo de actividades a realizar, los criterios para superar niveles, los datos a guardar, la graduación de actividades en diferentes niveles, etc.
- Diseño de una interfaz de usuario del juego, sin funcionalidad, para presentar a los responsables del entrenamiento de los usuarios, y así llegar a un diseño final de la interfaz.
- Estudio de las alternativas de implementación y elección de las tecnologías y herramientas involucradas.
- Implementación del juego, incluyendo verificaciones y pruebas del mismo.
- Evaluación con los usuarios finales.
- Análisis de los resultados del proyecto y planteamiento de posibles líneas de trabajo futuras.
- Elaboración de la memoria del proyecto.

1.5. ESTRUCTURA DE LA MEMORIA

La presente memoria se compone de los siguientes apartados:

- **Apartado 1. Introducción.** Es el apartado actual, contiene una breve descripción del problema y un análisis de las herramientas *software* que hay en el mercado. Se plantean los objetivos de este proyecto y la metodología de trabajo que se ha seguido.
- **Apartado 2. Diseño del juego.** Se expone un análisis de las características del colectivo al que va destinado y sus requisitos de accesibilidad y usabilidad. Además, se presenta el proceso seguido para realizar el diseño de la interfaz, así como una descripción en profundidad de la estructura y funcionamiento final del juego. Se finaliza señalando las adaptaciones realizadas en la interfaz y en los contenidos del juego.
- **Apartado 3. Implementación.** Contiene un estudio de todas las alternativas de implementación que se barajaron, así como una descripción de las herramientas y tecnologías empleadas en la implementación del juego. Se expone la organización de la aplicación junto con la funcionalidad de cada uno de los ficheros. Por último, se detallan los elementos que la conforman: base de datos e implementación de clases.
- **Apartado 4. Evaluación.** Este apartado contiene los detalles de la evaluación realizada del juego por profesionales docentes. Además, se incluye información sobre la utilización de la aplicación desarrollada por usuarios con síndrome de Down.
- **Apartado 5. Conclusiones y posibles trabajos futuros.** Este apartado presenta las conclusiones extraídas de la realización de este trabajo. Además, se citan algunos aspectos ampliables a la aplicación desarrollada, como puede ser añadir funcionalidad o puntos en los que se puede mejorar.

2. DISEÑO DEL JUEGO

2.1. CARACTERÍSTICAS DE LOS USUARIOS CON SÍNDROME DE DOWN

A la hora de diseñar *software* es esencial hacer un estudio previo de las necesidades del colectivo al que va dirigido. Cuando se plantea el diseño de esta aplicación para usuarios con limitaciones cognitivas dicha premisa se hace aún más importante para conseguir el desarrollo de un *software* adecuado.

En primer lugar se identifica el tipo de usuario al que va orientado el juego. En este caso son jóvenes con síndrome de Down que tienen un buen manejo del ordenador y que están en proceso de incorporación laboral en una gran ciudad como Madrid.

Ahora debemos sobre todo identificar las limitaciones cognitivas que presentan a la hora de jugar y de relacionarse con el entorno. Esto es fundamental para por ejemplo elegir los estímulos adecuados que permitan al jugador poder disfrutar del juego sin estar cohibido debido a sus limitaciones [11].

Muchos autores describen dificultades y necesidades de los usuarios de aplicaciones multimedia con síndrome de Down [5] [7] [14] [15]. Sin querer marcar a estos usuarios por el hecho de tener esta cromosomopatía y partiendo de la base que cada persona presenta unas características individuales, es conveniente tener en cuenta algunas dificultades y necesidades comunes de cara a planificar un *software* que cumpla con su objetivo y no limite sus posibilidades de aprendizaje. Estas características son las siguientes:

- **Dificultades de motivación y fatiga.** Cuando se encuentran ante actividades monótonas y/o con un grado de dificultad excesivo se puede originar un sentimiento de cansancio, frustración, sensación de fracaso sistemático, etc. que lleva al abandono de la tarea. Esta falta de motivación también se presenta por su dificultad para entender los objetivos de las tareas que se le proponen, por lo que no se enfrentan con deseo a las mismas.
- **Dificultad para mantener la atención.** Es una repercusión derivada de su inmadurez cerebral y que se manifiesta con una ausencia del interés suficiente para que se vea favorecido el aprendizaje. Esta dificultad conlleva que presenten problemas tanto de dispersión como de selección de estímulos.
- **Lentitud en los tiempos de reacción.** Es necesario que dispongan del tiempo necesario para que interpreten y respondan, no debiendo sustituirles ni apremiarles.
- **Dificultades en el procesamiento de la información visual y auditiva.** Especialmente presentan dificultad en el procesamiento de la información auditiva, prefiriendo el canal visual para recibir la información. Se debe a que suelen utilizar estrategias inadecuadas para analizar, asimilar e integrar la información.
- **Déficits con el lenguaje.** Esto es consecuencia del ya comentado procesamiento defectuoso de la información que reciben por vía auditiva, que conlleva que tengan problemas tanto en la comprensión como en la percepción del entorno.
- **Problemas en la memoria a corto plazo.** Se encuentra también relacionada con las dificultades auditivas y verbales, ya que se presenta sobre todo cuando la información se facilita por estos canales.

- **Dificultades en la memoria a largo plazo.** Cuando tienen que recuperar información acumulada en la memoria a largo plazo la realizan de manera lenta y con dificultades.
- **Dificultades en la comprensión de conceptos.** Por ejemplo tienen dificultades en la comprensión de los números (en ocasiones aprenden a contar si entenderlo).
- **Deficiencia en el cálculo aritmético.** Les cuesta mucho el aprendizaje de los cálculos más sencillos, así como imaginar soluciones a los pequeños problemas cotidianos.
- **Dificultad en la transferencia y consolidación de aprendizajes.** Se trata de un aspecto que hay que tener muy en cuenta en esta aplicación, ya que el objetivo es que puedan generalizar los conocimientos a contextos diferentes a los utilizados durante su proceso de adquisición.

Como ya se ha comentado en el capítulo anterior, algunas de las características de los usuarios con síndrome de Down pueden ser contrarrestadas con el uso de *software* adaptado, como instrumento de aprendizaje. Por ejemplo, permitiendo la adaptación de la presentación de la información (vocabulario, tamaño de letra, expresiones, mensajes de refuerzo positivo...) y del contenido a sus necesidades.

Una vez conocidas las características particulares de los usuarios a los que va destinado el juego estamos en posición de establecer una serie de requisitos relacionados con la usabilidad y la accesibilidad que debe cumplir el juego que se ha desarrollado.

2.2. REQUISITOS DE USABILIDAD Y ACCESIBILIDAD

En este apartado, se hace una enumeración de aquellos requisitos que se cree debe cumplir el juego, en base al análisis realizado en el apartado anterior, para que sea accesible por los usuarios. Estos requisitos se han sacado a partir de diferentes fuentes como las presentadas en [5] [7] [14] [15]. A continuación, se presentan estos requisitos junto con recomendaciones a tener en cuenta a la hora de diseñar el juego.

- Realizar un diseño de carácter lúdico y motivador, para conseguir un método de aprendizaje más atractivo para el usuario, intentando captar su atención con elementos motivantes para él, que fomenten su interés por la tarea.
- Cuidar la disposición, organización y presentación de los elementos en pantalla, de forma que ayuden a reforzar la comprensión del enunciado.
- Prestar atención al tamaño de gráficos y letra (tipo, tamaño y color que resalten del fondo), imágenes (tamaño, claras, con contornos nítidos, colores vivos, efecto de zoom, etc.) facilitando la lectura y comprensión.
- Diseñar una herramienta de fácil uso.
- Presentar la información relevante en el momento y la forma oportunos, para así ayudarles a focalizar su atención sobre los elementos relevantes de la tarea.
- Disminuir el número de factores distractores que dificulten la concentración en la tarea a realizar.

- Presentar las actividades secuenciadas según el grado de dificultad (de menor a mayor dificultad).
- Personalización para favorecer la adecuación del proceso al ritmo y estilo de aprendizaje de la persona.
- Utilizar la estrategia de repetición ya que es la más sencilla y la más usada para mejorar los resultados de la memoria. Para minimizar los problemas en la memoria de trabajo que presentan las personas con síndrome de Down se requiere el uso de materiales que puedan ser presentados de forma reiterada y flexible para facilitar su organización y elaboración.
- Proporcionar un mayor tiempo de aprendizaje (sin tiempo límite de respuesta) y realizar más práctica (número de repeticiones). Los aprendizajes se podrán repetir periódicamente para que así puedan ser conservados, no dándolos por definitivamente establecidos.
- Facilitar información adicional para realizar algunos ejercicios (ayudas individualizadas).
- Primar la presentación de la información por el código visual, frente al auditivo, dadas las dificultades que presentan de percepción por dicho canal.
- Presentar los problemas dentro de un contexto en situaciones cotidianas al alumno, para así facilitar la transferencia de conocimientos al mundo real.
- Fomentar los aprendizajes funcionales y que se apoyan en el contexto, para ayudar en la comprensión de conceptos (p.e. números).
- Utilizar reforzadores distribuidos de forma coherente y adecuada. Dar siempre refuerzos positivos al usuario que le motiven a continuar jugando.
- Utilizar refuerzos visuales complementarios para mitigar las dificultades de memoria a corto plazo.
- Adaptar la interacción y crear una interfaz accesible para solucionar los problemas que presentan de motricidad fina.
- Presentar estructuras dinámicas, elementos en movimiento, inclusión de ejemplificaciones y experiencias a la hora de abordar las dificultades en pensamiento abstracto se
- Proporcionar instrucciones claras y muy detalladas.
- Utilizar un lenguaje familiar adecuado al nivel cognitivo del usuario (especial consideración sobre la complejidad del vocabulario) tanto en los enunciados de las cuestiones, como en los mensajes de retroalimentación del *software*.
- Facilitar la máxima información posible de lo que debe hacer en cada momento.

A modo de conclusión, parece obvia la necesidad de tener en cuenta las características de las personas con Síndrome de Down, en especial aquellas que afectan a sus procesos de aprendizaje, para poder ofrecer una respuesta adecuada a sus necesidades.

2.3. ASPECTOS GENERALES DEL JUEGO

El aprendizaje se realiza de manera incremental, por ello el juego está estructurado en un sistema de niveles, en concreto 4 niveles de creciente dificultad. Cada uno de los niveles tiene una serie de cuestiones asociadas, que tienen que ser contestadas por los jugadores. El juego consiste en una serie de rondas de preguntas asociadas a cada nivel. Cuando un jugador selecciona jugar una ronda de un nivel concreto se le plantean 10 preguntas escogidas de entre las asociadas a ese nivel. Para poder tener acceso al nivel inmediatamente superior el jugador debe alcanzar un porcentaje mínimo de respuestas correctas, en concreto debe contestar correctamente al 80% de las preguntas planteadas para ese nivel. Una vez alcanzado ese hito tendrá acceso al siguiente nivel.

El juego permite al usuario jugar tantas rondas como desee de los niveles a los que tenga acceso en cada momento. El porcentaje de respuestas acertadas se calcula sobre el total de contestaciones dadas en todas las rondas que se hayan completado de cada nivel. El jugador puede acceder en todo momento a jugar rondas de niveles inferiores que ya haya superado. El objetivo es que el usuario pueda reforzar los conocimientos que previamente ha adquirido. Si al jugar una ronda de un nivel ya superado obtiene una puntuación para ese nivel menor al 80% se le desactiva el acceso a los niveles superiores. Las preguntas del juego pueden clasificarse inicialmente en dos grupos:

- El primer grupo está compuesto por preguntas que tienen el objetivo de conseguir que el jugador conozca y se familiarice con la red de metro de Madrid. Por ejemplo, preguntas tales como: *“¿Me puedes decir de qué color es la línea X de metro?”*, *“¿Sabes a qué dos líneas de metro pertenece la estación A?”*, *“¿Me puedes decir el nombre de la estación que falta en este dibujo?”*, etc.
- En el segundo grupo se encuentran aquellas preguntas que plantean situaciones reales, tanto cotidianas como extraordinarias a las que el jugador tiene que encontrar una solución. El objetivo de estas cuestiones es entrenar a los usuarios en el proceso de toma de decisiones. Algunos ejemplos son: *“Necesito ir desde Lavapiés, en la línea 3, hasta Méndez Álvaro, de la línea 6, ¿en qué estación tengo que hacer transbordo?”*, *“¿Puedo ir desde la estación Ibiza hasta la estación Vinateros sin hacer transbordo?”*, *“He cogido el metro en Ascao, línea 7, para ir a Cuatro Caminos. Iba a hacer transbordo a la línea 6 en Avda. de América, pero ha habido una avería. Si sigo el viaje en la línea 7, ¿hasta qué estación tendré que ir para coger la línea 2 y llegar a Cuatro Caminos?”*.

En general y en función del nivel, con cada pregunta se presentan dos o más posibles respuestas y el jugador tiene que seleccionar la correcta. En el nivel de mayor dificultad (*nivel 4*), hay preguntas con una o dos cajas de texto en las que el jugador tiene que escribir su/s respuesta/s a la pregunta. Cada pregunta tiene una única respuesta correcta asociada. Las respuestas a las cuestiones pueden ser atómicas o compuestas.

- Las preguntas de respuesta atómica son en las que el jugador tiene que seleccionar o escribir una única opción de entre las posibles respuestas que se le ofrecen y de las que sólo una es correcta (ver ejemplos mostrados en la Figura 2.1.).
- Las cuestiones de respuesta compuesta son aquellas en las que la respuesta se compone por el conjunto de dos o tres respuestas de entre las opciones de contestación que se presentan al jugador y sólo hay una combinación correcta (ver ejemplos mostrados en la Figura 2.2.).



Figura 2.1.- Ejemplos de preguntas de respuesta atómica



Figura 2.2.- Ejemplos de preguntas de respuesta compuesta

Cada vez que se contesta a una pregunta es necesario mostrar una pantalla de retroalimentación al jugador. En ella se incluye el enunciado de la pregunta planteada, la respuesta dada por el jugador y la respuesta correcta a la pregunta. En caso que el usuario haya contestado correctamente se incluye un mensaje de felicitación, y en todos los casos, un mensaje de motivación. En la Figura 2.3., se muestran dos ejemplos de retroalimentación.



Figura 2.3.- Ejemplos de retroalimentación dada al jugador

2.4. PROCESO DE DISEÑO DE LA INTERFAZ

El proceso de diseño de la estructura y la interfaz ha seguido una metodología iterativa, mediante la fabricación de prototipos utilizando la herramienta *Microsoft Office PowerPoint 2007* del paquete *Microsoft Office*.

Se ha contado con el asesoramiento, colaboración y validación final de los profesores que trabajan diariamente con personas con discapacidades intelectuales. A éstos se les ha presentado tanto la interfaz como la estructura propuestas y se han analizado en conjunto los posibles problemas, mejoras, cambios, etc. Basándose en estos hallazgos se ha modificado el prototipo realizado hasta obtener el definitivo. Se han contemplado en todo momento los requisitos de accesibilidad y usabilidad enunciados anteriormente. Además a la hora de presentar los prototipos a los profesores se acompañaron de un diagrama indicando la secuencia de pantallas del juego como el que sigue:

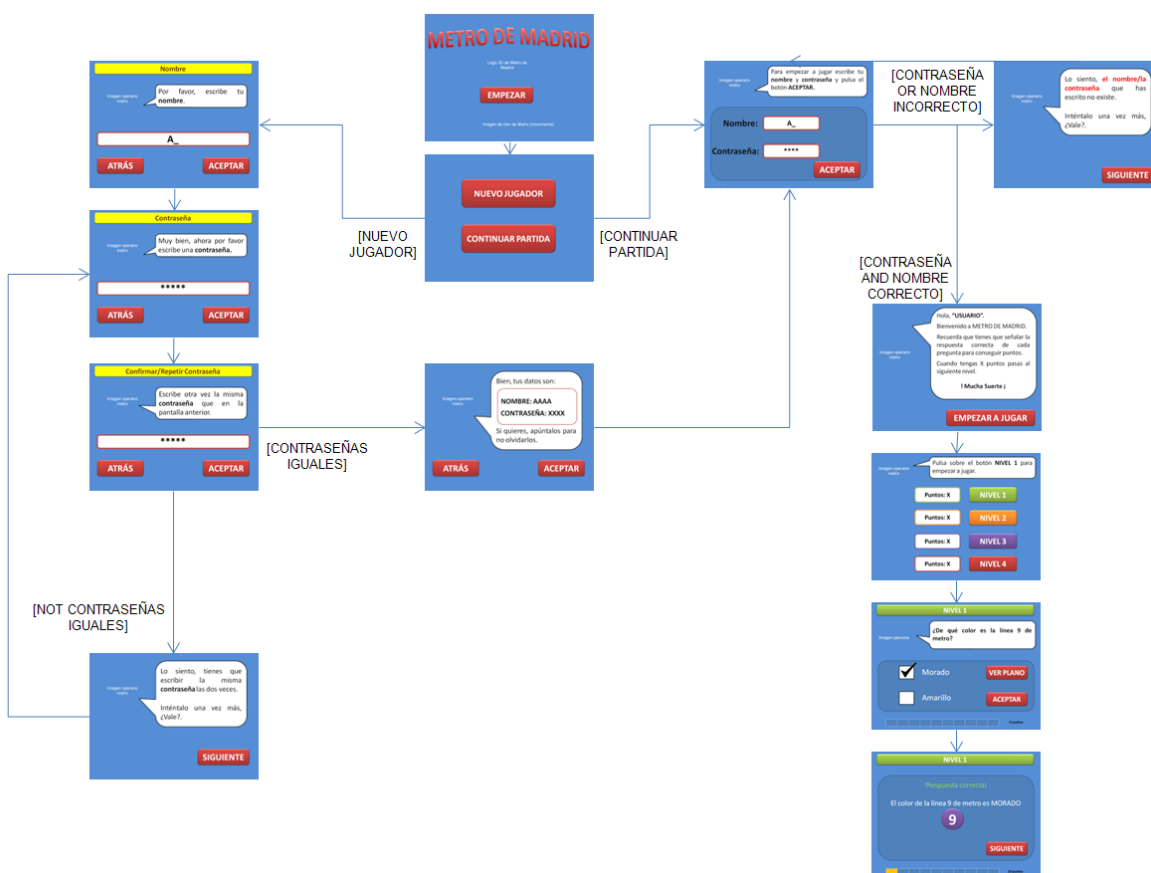


Figura 2.4.- Diagrama de pantallas del juego presentado a especialistas durante la fase de diseño.

A continuación se presentan los prototipos diseñados para las pantallas más significativas del juego, utilizados en la fase inicial de validación de la usabilidad y accesibilidad. Se indican los principales problemas detectados y las modificaciones realizadas hasta llegar a la versión final.

Tanto en la pantalla de pregunta como en la de respuesta se ofrece un ejemplo de las mismas. El proceso de diseño que se ha seguido con ambas es generalizable a las restantes pantallas de respuesta/pregunta del juego.

⇒ Pantallas iniciales del juego

El primer prototipo de las pantallas iniciales del juego, mostrado en la Figura 2.4., incluye, una pantalla de inicio del juego seguida de las pantallas necesarias para el registro de un nuevo usuario, las pantallas para identificar a un usuario ya existente que desea continuar jugando y una pantalla de bienvenida al juego una vez identificado el jugador correctamente.

La pantalla en la que el usuario debía elegir registrarse como nuevo usuario o continuar una partida si ya estaba registrado no era clara, por lo que se debía dar más información al jugador sobre lo que debía hacer en ese momento.

Se valoró la necesidad y ventajas de contar con contraseña para acceder al juego. Finalmente se descarta su uso por no ser considerada necesaria por los profesores debido a los problemas que puede ocasionar a los jugadores para acceder al mismo, derivados de las dificultades de memoria que presentan y que se han comentado anteriormente.

Desde un primer prototipo se fue evolucionando hasta llegar al prototipo final mostrado en la Figura 2.5. En este prototipo, se añade una breve pero necesaria explicación en la pantalla de elección entre registrarse o continuar jugando. Además, se suprimen todas las referencias a la contraseña. También, se hace una diferencia en la pantalla que se muestra tras el registro correcto en el juego, diferenciando el primer acceso al juego y accesos posteriores. Por último, se incluye el botón para salir del juego en la parte superior de las pantallas.

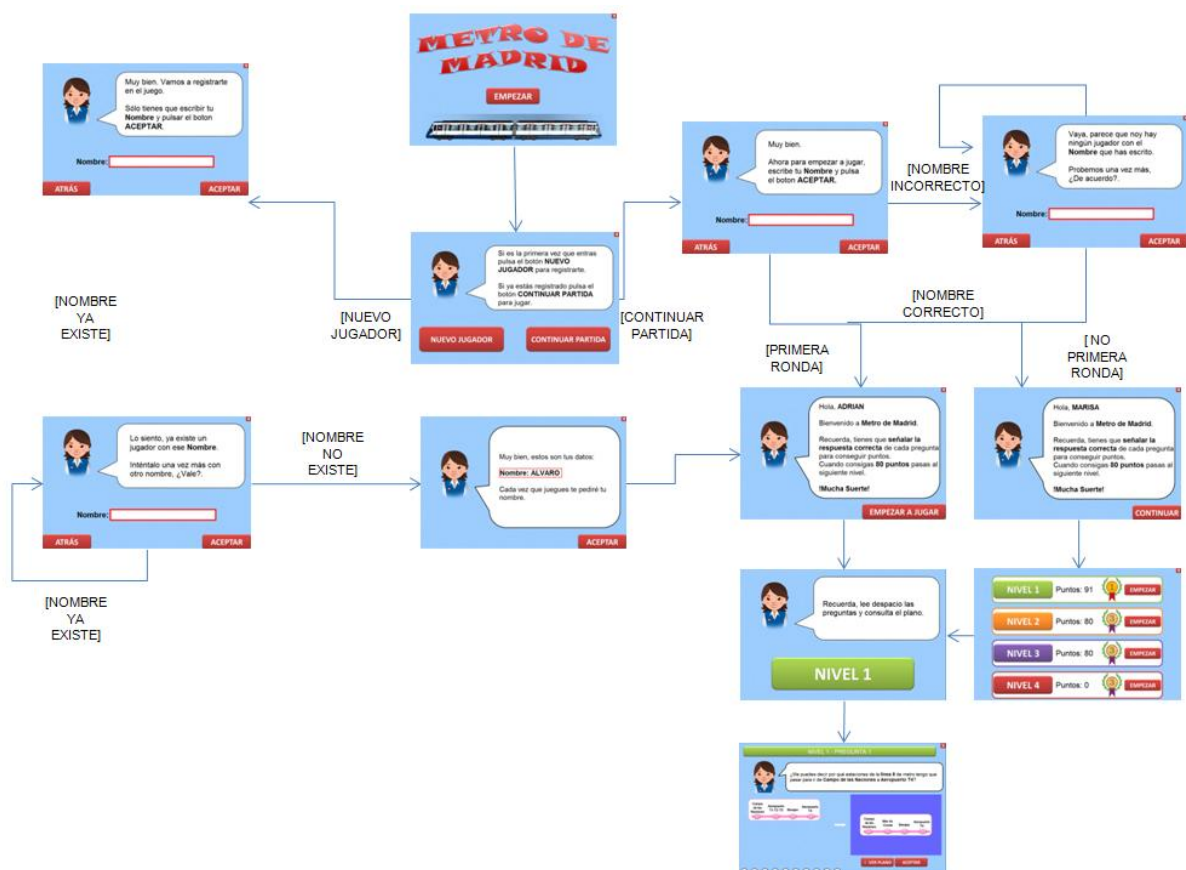


Figura 2.5.- Prototipo final de las pantallas iniciales

⇒ **Pantalla de pregunta**

El primer prototipo de las pantallas de pregunta fue el que se muestra en la Figura 2.6. Se puede observar que la manera de presentar el color de las líneas de metro está descontextualizada del plano real de Metro de Madrid, no se aprovecha al máximo el canal visual como elemento motivador. El enunciado de la pregunta es impersonal.



Figura 2.6.- Primer prototipo de la pantalla de pregunta

A partir del primer prototipo, se obtuvo el que se presenta en la Figura 2.7. En el segundo prototipo se representan los símbolos de la línea de metro tal y como éstos aparecen en el contexto del plano del Metro de Madrid. Se hace uso del logo de Metro de Madrid para ayudar a la localización e identificación del botón “Ver plano”. Se diseña un área de respuesta del jugador para separar así la respuesta dada de todas las posibles respuestas. Por último, se añade información sobre el número de pregunta en la que se encuentra el jugador para favorecer así la orientación en el juego.

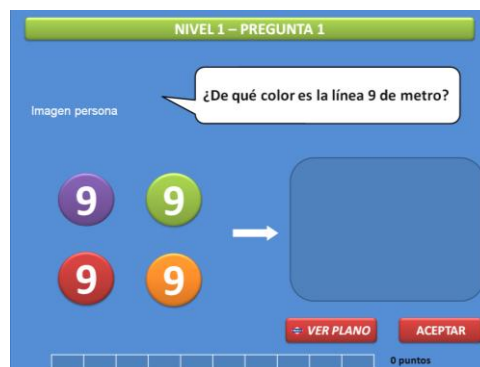


Figura 2.7.- Segundo prototipo de la pantalla de pregunta

Finalmente, se obtuvo el prototipo final presentado en la Figura 2.8., en el que se añade el botón para salir del juego en la parte superior derecha, y se adecua el número de posibles respuestas dependiendo del nivel de juego. En el ejemplo de la Figura 2.8. se muestran sólo dos posibles respuestas al ser una pregunta del primer nivel. Se resalta en negrita partes importantes del enunciado (p.e. números de línea de metro, nombre de estaciones, entre otros). Se sustituye la barra de progreso donde se mostraba la puntuación del jugador por 10 iconos que representan las respuestas da el jugador a las 10 preguntas que componen la ronda que está jugando. En la Figura 2.8 se puede observar que el jugador en la ronda actual ha contestado correctamente a las preguntas 3, 5 y 7, erróneamente a las preguntas 1, 2, 4, 6 y 8 y le faltan por contestar 2 preguntas para completar la ronda.



Figura 2.8.- Prototipo final de la pantalla de pregunta

⇒ **Pantalla de respuesta**

Respecto a las pantallas de respuesta, se puede ver el primer prototipo en la Figura 2.9. En esta pantalla se puede observar que la retroalimentación que se presenta al jugador es negativa. No se recuerda ni el enunciado de la pregunta, ni la respuesta dada por el jugador. Así que no se puede hacer una comparación visual y rápida de las respuestas junto con el enunciado. Además la información se presenta de manera impersonal.

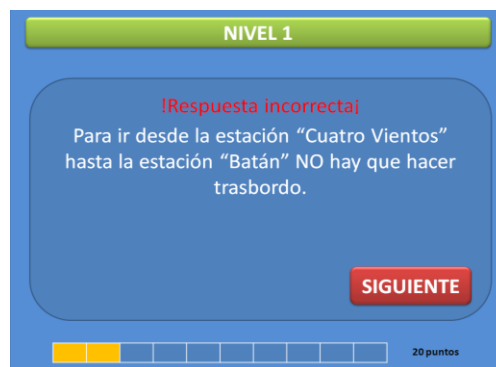


Figura 2.9.- Primer prototipo de la pantalla de respuesta

En el segundo prototipo (ver Figura 2.10.), el *feedback* que se da al jugador sigue siendo negativo. Se presenta tanto la respuesta dada a la pregunta, como la respuesta correcta pero sigue sin mostrarse el enunciado de la pregunta. Se añade información sobre el número de pregunta actual. Se personaliza la retroalimentación añadiendo al personaje del juego.

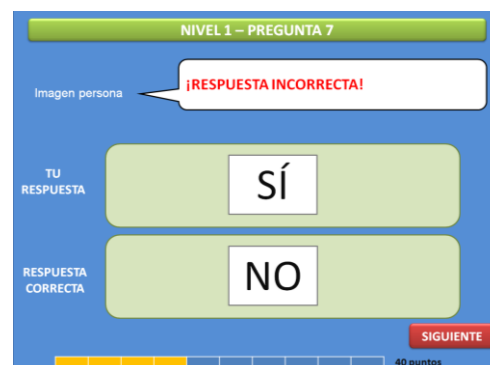


Figura 2.10.- Segundo prototipo de la pantalla de respuesta

El prototipo final se puede ver en la Figura 2.11. El *feedback* siempre es positivo, animando al usuario a seguir jugando. Se añade el enunciado de la pregunta para que tenga la información completa y un botón que permite salir del juego en la parte superior derecha.

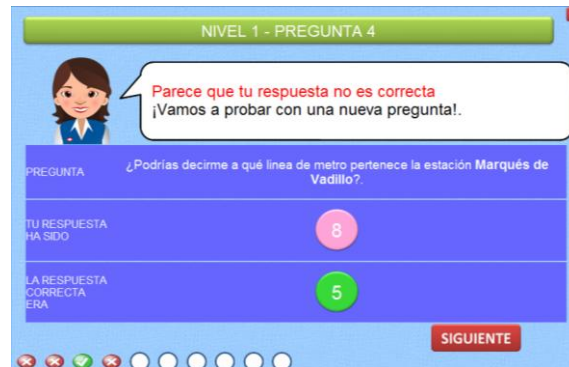


Figura 2.11.- Prototipo final de la pantalla de respuesta

⇒ **Pantalla de resumen de puntuaciones**

Se hicieron varios prototipos de la pantalla de resumen de puntuaciones. El prototipo final se presenta en la Figura 2.13. En este prototipo, se tiene el botón “Empezar” en cada nivel. Se opta por incluir como representación gráfica de la puntuación unas medallas. Además se incluye el botón para salir del juego.

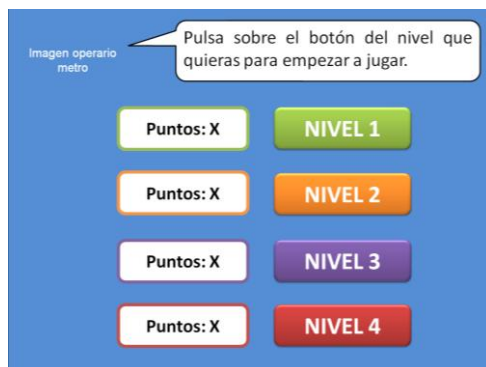


Figura 2.12.- Primer prototipo de la pantalla de resumen de puntuaciones



Figura 2.13.- Prototipo final de la pantalla de resumen de puntuaciones

Como fuentes para los textos que están presentes en el juego se han utilizado “arial”, “courier new” e “impact” tanto por su aprobación por parte de las personas encargadas del entrenamiento como por su fácil disponibilidad.

2.5. ESTRUCTURA Y FUNCIONAMIENTO DEL JUEGO

El juego comienza con la pantalla de inicio que incluye en la parte superior el nombre del mismo, en la zona central el botón “Empezar” y en la región inferior una imagen de un metro. Cuando el usuario pulsa sobre el botón “Empezar” da comienzo el juego. A continuación, se muestra una pantalla donde el jugador deberá indicar si ya está registrado en el sistema o si es un nuevo jugador.

Si es un nuevo jugador, se le pide el nombre con el que quiere identificarse a partir de ese momento para poder acceder al juego (ver Figura 2.14.).



Figura 2.14.- Pantalla de registro de nuevos usuarios

Una vez que escribe el nombre, si el nombre ya existe en la base de datos de jugadores se le informa y se le solicita que introduzca otro nombre (ver parte izquierda de la Figura 2.15.). En caso contrario, se muestra una pantalla confirmando el registro correcto y mostrando el nombre con el que tendrá que entrar en el juego cada vez que quiera jugar (ver parte derecha de la Figura 2.15.).



Figura 2.15.- Pantallas de error y confirmación de registro de un nuevo usuario

Si se trata de un jugador ya registrado en el juego se le solicita el nombre para iniciar el juego. En caso de introducir un nombre que no se encuentre en la base de datos se le solicita de nuevo que introduzca el nombre.



Figura 2.16.- Pantallas de registro de usuario ya existente y de error de registro

Una vez que el usuario está validado o registrado en el sistema, si no ha finalizado nunca una ronda del primer nivel, en la página de bienvenida se muestra el botón "Empezar a jugar" (ver parte izquierda de la Figura 2.17.). Cuando el usuario pulse sobre él, iniciará una

ronda del juego en el *nivel 1*. Si el jugador ya había jugado antes el texto del botón es “Continuar” y tras pulsarlo se muestra la página resumen con los puntos que ha obtenido en cada nivel (ver Figura 2.18.).

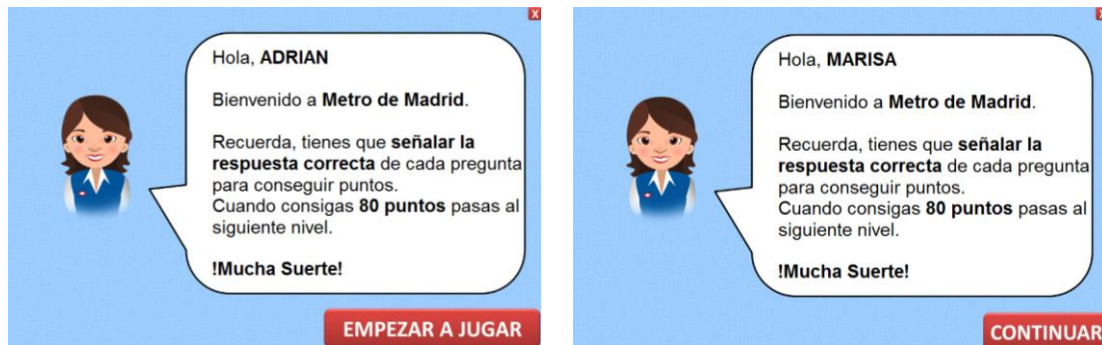


Figura 2.17.- Pantallas de bienvenida.

En la pantalla de resumen de puntuaciones el jugador puede consultar sus avances a largo del juego. Esta pantalla tiene una fila por cada nivel al que tiene acceso el jugador en ese instante. Cada fila está dividida en 4 áreas:

NIVEL 1	Puntos: 95		EMPEZAR
NIVEL 2	Puntos: 90		EMPEZAR
NIVEL 3	Puntos: 90		EMPEZAR
NIVEL 4	Puntos: 0		EMPEZAR

Figura 2.18.- Estructura de la pantalla de resumen de puntuaciones

- ① **NIVEL.-** Nombre del nivel al que corresponde la información contenida en la fila.
- ② **PUNTUACIÓN.-** Contiene la puntuación, hasta el momento, del jugador en el nivel correspondiente (número comprendido entre 0 y 100). La puntuación que se muestra tiene en cuenta todas las respuestas dadas por el jugador a las preguntas de cada nivel. Por ejemplo en la Figura 2.18. en el *nivel 3* se indica que el usuario tiene 90 puntos. Esto quiere decir que, de todas las cuestiones respondidas en diferentes rondas de este nivel por el jugador, el 90% de las respuestas han sido correctas.
- ③ **REPRESENTACIÓN GRÁFICA DE LA PUNTUACIÓN.-** A continuación de los puntos obtenidos por el usuario en cada nivel, hay un dibujo de una medalla. Ésta puede ser de oro, plata o bronce. Oro representa que el jugador ha respondido más del 90% de las preguntas correctamente; plata significa que ha conseguido un porcentaje de aciertos entre 80-90% ambos incluidos; y bronce representa menos del 80% de respuestas acertadas.

- ④ **BOTÓN EMPEZAR.**- Al pulsar sobre el botón empezar se comienza una ronda de preguntas del nivel elegido.

Cuando el jugador pulsa sobre el botón “Empezar” del nivel en el que quiere jugar, el juego empieza, presentando una ronda de 10 preguntas correspondientes al nivel seleccionado. Antes de comenzar con las preguntas de la ronda que corresponda se muestra una pantalla con un recordatorio para el jugador para que lea despacio las preguntas que se le plantean y animándole a consultar el plano de metro para resolver las cuestiones. Además, se muestra un botón con el nivel seleccionado.

Si el jugador pulsa sobre el botón, se le muestra la primera pregunta de la ronda. Si transcurren 10 segundos sin que presione el botón, el programa le muestra la primera pregunta automáticamente.

Cada pregunta se presenta en una pantalla que está compuesta de 5 áreas para la presentación de la información (ver Figura 2.19.).

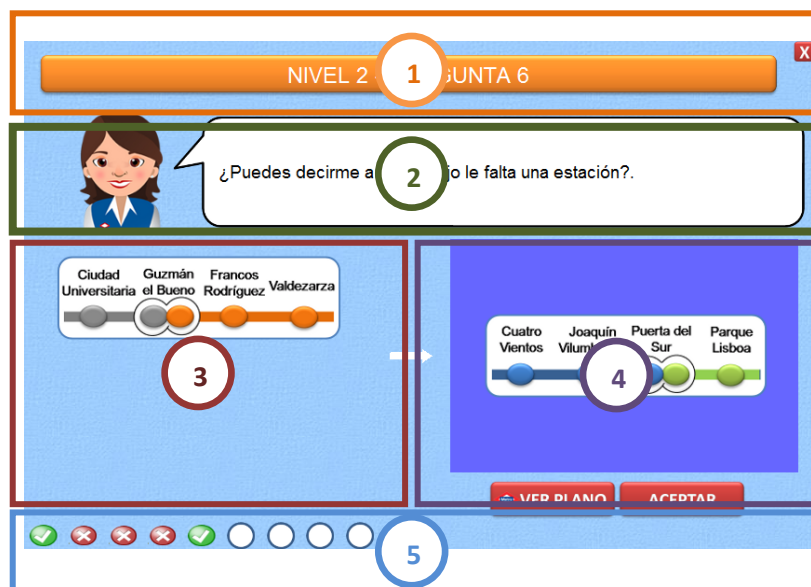


Figura 2.19.- Estructura de la pantalla de pregunta

En el área ①; se muestra un encabezado que contiene información sobre el nivel y el número de pregunta actual junto con el botón que permite salir del juego. El enunciado de la pregunta planteada se muestra en el área ②. Debajo del enunciado se sitúan las respuestas posibles (área ③) y la respuesta del jugador (área ④). En el área con las respuestas posibles se muestran dos, tres ó cuatro opciones a la pregunta. En función del nivel y de la pregunta el jugador deberá seleccionar una, dos o tres respuestas. En el área coloreada en la parte derecha de la pantalla está destinada a contener la respuesta del usuario. Para situar la respuesta sobre esta área sólo es necesario que pulse sobre la respuesta que desea del área de respuestas posibles. Una vez colocadas en esta área, las respuestas pueden ser fácilmente reemplazadas por otras, con el mismo procedimiento anterior (pulsando sobre la respuesta deseada). En la zona inferior del área de respuestas del jugador hay dos botones: “Ver plano” que abre una ventana con un plano de la red de metro de Madrid y el botón “Aceptar” para comprobar si la respuesta dada es correcta.

El usuario siempre que lo necesite podrá acceder al plano del Metro. El juego no penaliza que el usuario consulte el plano ya que lo importante en este caso, es que sepa interpretarlo para poder ir contestando las preguntas que el juego le va realizando.

Por último, en la parte inferior (área ⑤ de la Figura 2.19.) se muestra la puntuación del jugador sobre la ronda actual. Se compone de 10 iconos (uno por cada pregunta de la ronda). Al comenzar la ronda, los 10 iconos son un círculo blanco. Los círculos blancos representan las preguntas que faltan por contestar de la ronda. Según el jugador va contestando preguntas, estos círculos van cambiando por un icono verde con un "tick" que representa una respuesta correcta o por un icono rojo con un aspa que representa una respuesta incorrecta. Por ejemplo, en la Figura 2.19 el jugador ha contestado 5 preguntas de la ronda, habiendo contestado correctamente la primera y la quinta y de manera errónea las restantes, además vemos que le faltan 5 preguntas por contestar para completar la ronda.

Hay algunas excepciones sobre este diseño estándar como las preguntas donde se pide al jugador que escriba a través del teclado su respuesta/s en caja/s de texto. Cuando la respuesta del usuario es una cadena de caracteres y su respuesta difiere en 3 o menos caracteres de la respuesta correcta, falta una letra, hay espacios en blanco excesivos o en lugares incorrectos, o hay errores gramaticales (tildes, mayúsculas, minúsculas) se muestra una pantalla en la que se da la oportunidad al jugador de rectificar la respuesta dada por la que corresponde (ver Figura 2.20.).



Figura 2.20.- Pantallas de pregunta del tipo 1 del nivel 4 y su posibilidad de rectificación.

⇒ Tipos de preguntas

Como ya se ha comentado, las preguntas están agrupadas en el juego en 4 niveles de dificultad. A su vez hay 6 tipos de preguntas diferentes dentro de las cuestiones que componen cada nivel. Cada uno de estos 6 tipos hace preguntas sobre distintos aspectos que el jugador debe dominar, como por ejemplo: número de estaciones de un trayecto, color de las líneas de metro, líneas a las que pertenece cada estación, estaciones en las que hay que hacer transbordo en un trayecto para recorrer el número mínimo de estaciones posible, etc. A continuación se explican los seis tipos de preguntas que contempla la aplicación:

- **TIPO 1.-** En todos los niveles se plantea la misma pregunta: "¿Me puedes decir el nombre de la estación que falta en el dibujo?". Esta pregunta se encuentra acompañada de un fragmento del plano de Metro de Madrid al que le faltan una o dos estaciones.

En los niveles 1 y 2 se presentan imágenes de trayectos sin transbordo. La graduación de dificultad viene dada porque en las imágenes del nivel 1 todas las estaciones pertenecen únicamente a la línea del trayecto y en el nivel 2 hay alguna estación que pertenece, además a otra línea. En ambos niveles debe seleccionar una única respuesta posible, ofreciendo el juego dos respuestas posibles en el nivel 1 y tres en el nivel 2 (ver Figura 2.21 y Figura 2.22).

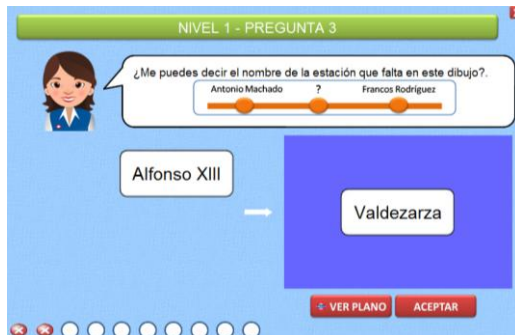


Figura 2.21.- Pregunta tipo 1 del nivel 1

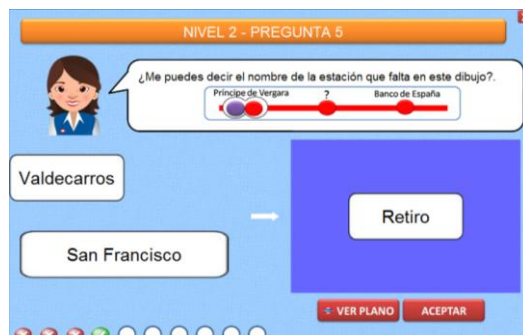


Figura 2.22.- Pregunta tipo 1 del nivel 2

En el nivel 3 las imágenes son de trayectos con un transbordo y el jugador debe seleccionar una de las tres respuestas posibles que ofrece el juego (ver Figura 2.23.). Por último, las preguntas del nivel 4 se refieren a trayectos con dos transbordos. Las imágenes muestran trayectos a los que les faltan dos estaciones, de las que el jugador debe escribir el nombre en las cajas de texto correspondientes (ver Figura 2.24.).

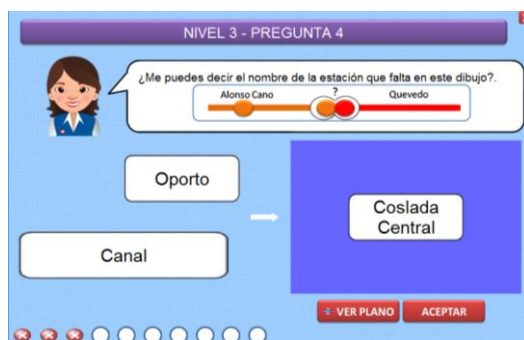


Figura 2.23.- Pregunta tipo 1 del nivel 3

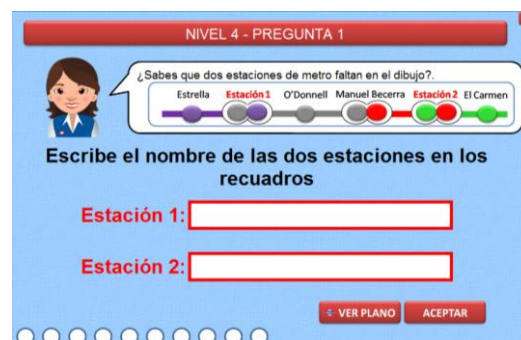


Figura 2.24.- Pregunta tipo 1 del nivel 4

- **TIPO 2.-** En este tipo de preguntas, el jugador tiene que responder a preguntas relacionadas con una estación origen y otra estación destino. De un nivel a otro varía la pregunta que se plantea.

Las preguntas del nivel 1 se refieren a trayectos sin transbordo donde el jugador debe seleccionar el dibujo correcto de los dos que el juego le ofrece como respuesta posible. La estructura de las preguntas de este nivel es: "¿Me puedes decir por qué estaciones de la línea "x" de metro tengo que pasar para ir de "a" a "b"?" (ver Figura 2.25.).

En el nivel 2 las preguntas que se plantean hacen referencia a trayectos con un transbordo y tienen la estructura "Necesito ir desde "a", en la línea "x", hasta "b", de la línea "y". ¿En qué estación tengo que hacer transbordo?". Al jugador se le ofrecen tres respuestas posibles de las que debe seleccionar una (ver Figura 2.26.).

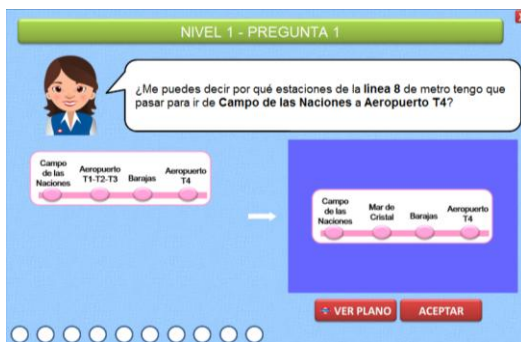


Figura 2.25.- Pregunta tipo 2 del nivel 1

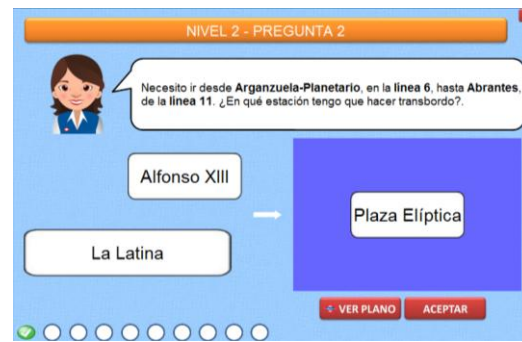


Figura 2.26.- Pregunta tipo 2 del nivel 2

En el nivel 3 se incrementa el número de transbordos a dos y el jugador debe seleccionar dos de las tres posibles respuestas que le ofrece el juego. Las preguntas que se plantean en este nivel tienen la estructura "He ido desde "a", de la línea "x", hasta "b", de la línea "y", si he recorrido en total "n" estaciones. ¿En qué dos estaciones he hecho transbordo?" (ver Figura 2.27.).

Las preguntas del nivel 4 plantean diferentes situaciones inesperadas que pueden pasar antes o durante un trayecto de metro. Explican una situación en la que se suspende parte de una línea de metro que impide realizar el transbordo deseado y se pregunta sobre en qué otra estación puede hacer un transbordo que le permita llegar a su destino. Debe seleccionar una de las tres posibles respuestas que le ofrece el juego (ver Figura 2.28.).

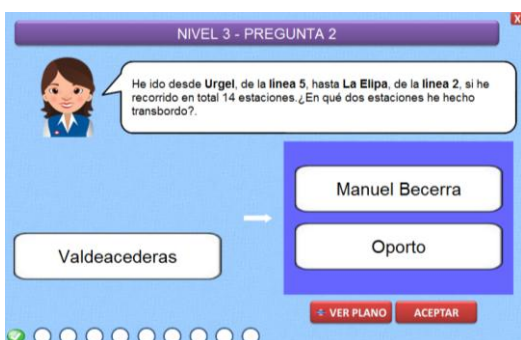


Figura 2.27.- Pregunta tipo 2 del nivel 3

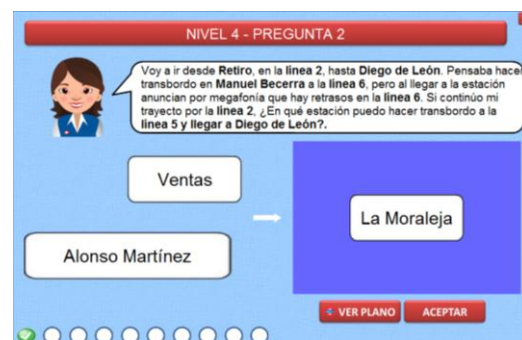


Figura 2.28.- Pregunta tipo 2 del nivel 4

- **TIPO 3.-** Este tipo de preguntas están en los niveles 1, 2 y 3. En los 3 niveles se plantea la misma pregunta: *¿Podrías decirme a qué línea/s de metro pertenece la estación "a"?*

La dificultad se va incrementando de manera que en el nivel 1 las estaciones pertenecen a una única línea de metro, en el nivel 2 son estaciones que pertenecen a dos líneas de metro y en el nivel 3 pertenecen a tres líneas de metro. Por ello en el nivel 1 sólo debe seleccionar una respuesta de las dos posibles que le ofrece el juego, en el nivel 2 debe seleccionar dos de las tres posibles respuestas que le ofrece el juego y en el nivel 3 debe seleccionar tres de las cuatro posibles respuestas que le ofrece el juego (ver Figura 2.29. y Figura 2.30.).

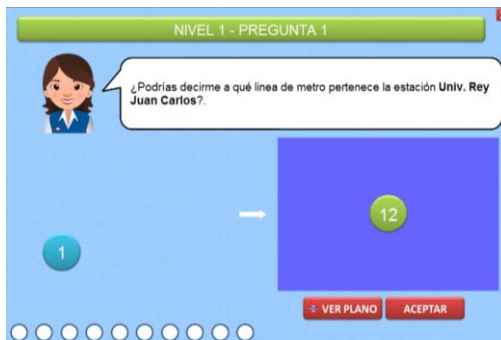


Figura 2.29.- Pregunta tipo 3 del nivel 1

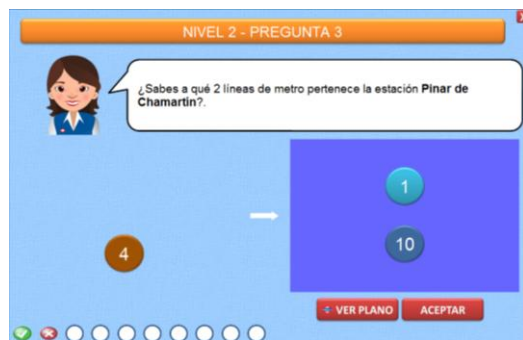


Figura 2.30.- Pregunta tipo 3 del nivel 2

- **TIPO 4.-** Este tipo de preguntas están en los niveles 1, 2 y 3. En los 3 niveles se plantea la misma pregunta: *¿Sabes decirme a qué dibujo le falta una estación?*

La dificultad se va incrementando según el jugador avanza de nivel. Así en las preguntas del nivel 1 se muestran imágenes de trayectos sin transbordo donde todas las estaciones pertenecen únicamente a la línea del trayecto, en el nivel 2 los trayectos mostrados tienen un transbordo y en el nivel 3 los trayectos son de dos transbordos. En todos los niveles el jugador debe seleccionar una de las dos respuestas posibles que le ofrece el juego (ver Figura 2.31.).

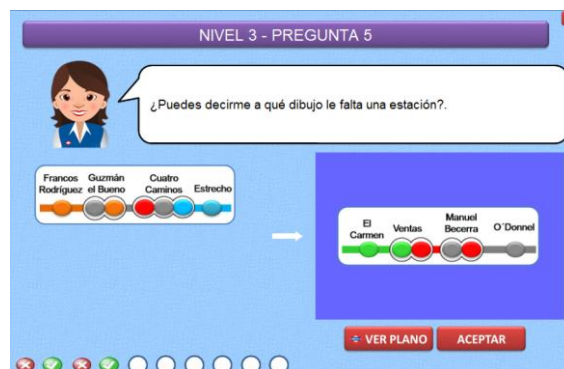


Figura 2.31.- Pregunta tipo 4 del nivel 3

- **TIPO 5.-** De un nivel a otro varía la pregunta que se plantea.

En el nivel 1 las preguntas que se plantean tienen la estructura *¿Me puedes decir de qué color es la línea "x" de metro?* y el jugador debe seleccionar una de las dos posibles respuestas que le ofrece el juego (ver Figura 2.32.).



Figura 2.32.- Pregunta tipo 5 del nivel 1

En el nivel 2 se plantean preguntas con la estructura “¿Puedo ir desde la estación “a” hasta la estación “b” sin hacer transbordo?”. El jugador debe seleccionar una de las dos posibles respuestas que le ofrece el juego (ver Figura 2.33.).

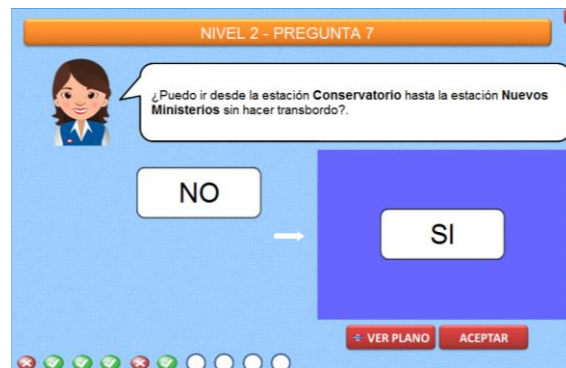


Figura 2.33.- Pregunta tipo 5 del nivel 2

En los niveles 3 y 4 se plantean diferentes situaciones inesperadas que pueden pasar antes o durante un trayecto de metro. Los enunciados explican una situación en la que se suspende una línea de metro que impide realizar el transbordo deseado y se pregunta sobre a qué otra línea puede hacerse un transbordo que permita llegar al destino. La graduación entre ambos niveles viene dada porque en el nivel 3 al jugador se le plantean tres posibles respuestas de las que debe seleccionar una (ver Figura 2.34.) y en el nivel 4 debe escribir en una caja de texto el número de la línea (ver Figura 2.35.).

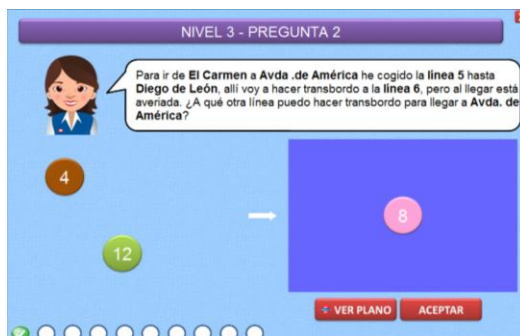


Figura 2.34.- Pregunta tipo 5 del nivel 3

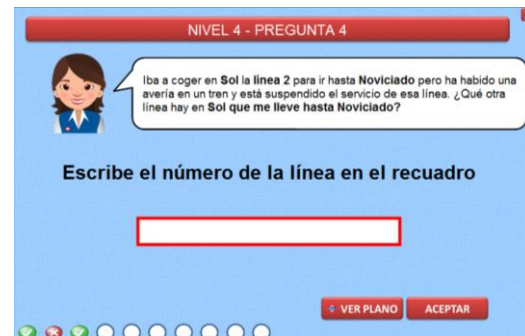


Figura 2.35.- Pregunta tipo 5 del nivel 4

- **TIPO 6.-** En los 4 niveles se plantean preguntas con la misma estructura, se plantea un trayecto con una estación origen y una estación destino y se pregunta sobre el número de estaciones que hay entre ambas.

En los niveles 1, 2 y 3 la primera parte de las preguntas planteadas tienen la misma estructura “Estoy en “a”, de la línea “x”, y voy hasta “b”, en la línea “y”, para hacer unas compras.” Difiere entre los 3 niveles la segunda parte de la pregunta planteada siendo “¿Cuántas estaciones hay sin hacer transbordo?” para el nivel 1, “¿Cuántas estaciones hay haciendo un transbordo?” para el nivel 2 (ver Figura 2.36) y “¿Cuántas estaciones hay haciendo dos transbordos?” para el nivel 3. Esto es así porque se ha graduado de manera que en el nivel 1 los trayectos que se plantean son sin transbordo y se va incrementando el número de transbordos según se avanza de nivel. Además en el nivel 1 al jugador se le ofrecen dos posibles respuestas posibles y en los niveles 2 y 3 son tres las posibles respuestas que ofrece el juego. En los tres niveles el jugador debe seleccionar una sola respuesta.

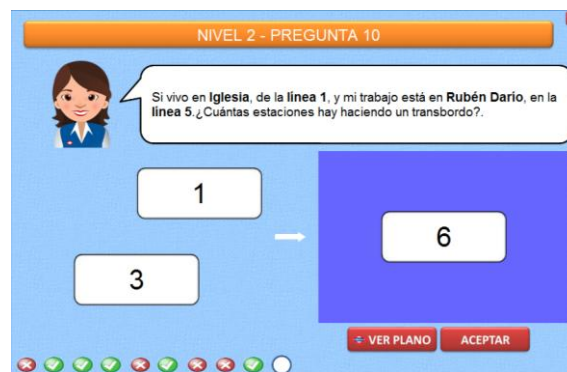


Figura 2.36.- Pregunta tipo 6 del nivel 2

En el nivel 4 se plantean preguntas con una estructura similar a: "He quedado con un amigo en "a", de la línea "x", para ir en metro al cine que está en "b", en la línea "y". ¿Cuántas estaciones hay por el camino más corto?" (ver Figura 2.37.). Son trayectos con n transbordos y el jugador debe escribir en la caja de texto el número de estaciones.



Figura 2.37.- Pregunta tipo 6 del nivel 4

A continuación, se explica detalladamente la dinámica del juego.

⇒ **Dinámica del juego**

El juego plantea al jugador una pregunta del tipo con menor número de aciertos. Así se garantiza que el jugador contesta correctamente a un porcentaje mínimo del 80% de todos los tipos del nivel antes de tener acceso a jugar al siguiente nivel. Si no fuese así y se eligiera de forma aleatoria se podría dar la situación que en la primera ronda de un nivel el jugador "tuviese suerte" y un alto porcentaje de las preguntas que se le planteasen correspondiesen a un tipo de preguntas que le resultaran fáciles y alcanzase el porcentaje que le permitiera pasar al siguiente nivel. No se aseguraría el dominio de todos los tipos de preguntas que componen el nivel.

Como se ha podido observar en los tipos de preguntas que utiliza el juego, en el diseño se han empleado algunas plantillas para crear preguntas. Así se pueden generar una gran cantidad de preguntas diferentes y no es necesario crear todos los contenidos para cada una de ellas. Con esto, se consigue aportar una mayor variedad en el repertorio de cuestiones del juego. En concreto, la Figura 2.38. muestra las plantillas generadas para los siguientes tipos de preguntas de cada nivel:

	NIVEL 1	NIVEL 2	NIVEL 3	NIVEL 4
TIPO 1	✗	✗	✗	✗
TIPO 2	✗	✓	✓	✗
TIPO 3	✓	✓	✓	
TIPO 4	✗	✗	✗	
TIPO 5	✓	✓	✗	✗
TIPO 6	✓	✓	✓	✓

Figura 2.38.- Tipos de preguntas generados mediante plantilla.

Además, en todas las preguntas, excepto las del *tipo 2* del *nivel 1*, debido a que las posibles respuestas son imágenes donde la respuesta correcta va asociada con la incorrecta, las respuestas falsas que se le ofrecen al jugador se generan mediante plantilla.

Cuando el jugador quiere comprobar si la respuesta dada es correcta, pulsa sobre el botón “Aceptar” situado debajo del área de respuestas dadas por el usuario. Si ha seleccionado un número inferior de respuestas a las esperadas se le vuelve a mostrar la misma página de pregunta con un mensaje recordándole el número de respuestas que debe seleccionar para la pregunta actual, hasta que el número de respuestas seleccionadas sea el correcto.

En la Figura 2.39., se muestra un ejemplo del *feedback* proporcionado por la aplicación al jugador relacionado con la respuesta dada. Como se puede observar, tiene una estructura similar a la pantalla de preguntas, en concreto consta de las siguientes áreas:

- ① **ENCABEZADO.**- Idem que en la pantalla de preguntas.
- ② **RETROALIMENTACIÓN.**- Contiene un mensaje escrito formado por dos frases. La primera da información sobre si la respuesta ha sido correcta o incorrecta. Si estamos en el primer caso el texto es de color verde y en el segundo de color rojo. La segunda frase es un mensaje de ánimo que motiva a continuar jugando.
- ③ **PREGUNTA.**- Muestra enunciado de la pregunta actual.
- ④ **COMPARACIÓN DE RESPUESTAS.**- Se muestra en primer lugar la respuesta dada por el usuario y a continuación la respuesta correcta a la pregunta, para que el jugador pueda comparar ambas. Cuando la respuesta está compuesta por más de una, en el caso de haber alguna coincidencia entre las respuestas dadas y las respuestas que forman la respuesta correcta se muestra de manera que queden alineadas las respuestas coincidentes, para así facilitar la comparación de respuestas.
- ⑤ **PUNTUACIÓN DEL NIVEL ACTUAL.**- Idem que en la pantalla de preguntas. Cuando el usuario pulse sobre el botón “Siguiente”, se le mostrará la siguiente pregunta o la pantalla que indica el final de la ronda que está jugando si ya ha contestado a las 10 preguntas de la misma.



Figura 2.39.- Estructura de la pantalla de respuesta

En el caso que el usuario acabe de contestar a la décima y última pregunta de una ronda del juego se muestra una pantalla de fin de nivel. Esta pantalla contiene la puntuación acumulada de todas las rondas que el usuario ha jugado en el nivel que acaba de finalizar, el dibujo de la medalla que corresponde con la puntuación y un mensaje de ánimo o recomendación que varía en función de la puntuación que haya obtenido (ver Figura 2.40.).



Figura 2.40.- Pantallas de fin de nivel

Una vez el jugador ha leído la información que se le muestra y pulsa sobre el botón “Siguiete” se le muestra la pantalla con el resumen de puntuaciones (ver Figura 2.41.). Esta pantalla contiene todos los puntos incluyendo los de la última ronda. En caso de haber superado los 80 puntos en el nivel más alto al que tenía acceso, se da acceso al nivel inmediatamente superior para que pueda también jugar a ese nivel, además de a todos los anteriores que ha superado. Por ejemplo, si el jugador obtiene 86 puntos en el nivel 3, la aplicación le habilita el nivel 4.

Para salir del juego el jugador tiene que pulsar el botón con un dibujo de “x” en su interior situado en la parte superior derecha de todas las pantallas. Tras pulsar en el botón se le pide confirmación para el abandono del juego. Además en el caso de estar jugando una ronda de preguntas se advierte que si abandona en ese momento el juego no se guardan los resultados de las preguntas de esa ronda que haya contestado hasta el momento. Es decir, si está en la pantalla de respuesta de la quinta pregunta de la segunda ronda que juega del nivel 3 y abandona en ese momento el juego sólo se guarda las puntuaciones de la primera ronda que jugó.



Figura 2.41.- Pantalla de resumen de puntuaciones

2.6. ADAPTACIONES

Para que el juego esté adaptado a las características ya analizadas de los usuarios con síndrome de Down se han realizado una serie de adaptaciones, tanto de contenido, como de presentación, funcionamiento, etc.

El diseño de la interfaz del juego se ha realizado de manera que su uso sea sencillo y al mismo tiempo resulte motivadora para el jugador. Se ha adaptado la interacción del usuario con el juego para que resulte accesible teniendo en cuenta las dificultades de motricidad fina que presentan los jugadores.

El lenguaje que se ha utilizado en la redacción de los textos que aparecen en el juego es adecuado al nivel cognitivo de los usuarios. Se proporcionan instrucciones claras, mediante frases cortas que dan información sobre la acción concreta que debe realizar el jugador en cada momento. Para facilitar la comprensión de los enunciados se han incluido elementos gráficos (fragmentos del plano de metro, iconos de líneas de metro, etc.) y se han dispuesto los elementos en las diferentes pantallas de forma estructurada. Se ha utilizado el tipo, color y tamaño de letra acordado con los profesores de los usuarios. Se ha optado por la presentación de la información siempre por el canal visual en detrimento del canal auditivo ya que el colectivo al que va dirigido el juego presenta dificultades en dicho canal.

Para ayudar a la focalización en los elementos relevantes en cada instante del juego se presenta en cada pantalla la información importante en cada momento, disminuyendo los distractores que disminuyen la concentración en la tarea.

Las preguntas se encuentran agrupadas en niveles de menor a mayor dificultad. En algunas preguntas se han incluido ayudas, en forma de elementos gráficos, para facilitar la respuesta de dichas preguntas. Se han incluido preguntas que reflejan situaciones cotidianas, para ayudar a que se produzca una transferencia de los conocimientos adquiridos mediante el uso del juego a las situaciones del mundo real.

Se ha dado mucha importancia a la fórmula de la repetición en el diseño de la estructura del juego para así mitigar los problemas de memoria que presentan las personas con síndrome de Down. No se ha introducido tiempo límite de respuesta.

Los refuerzos que se dan al jugador son siempre positivos y se encuentran distribuidos en distintas pantallas del juego.

3. IMPLEMENTACIÓN

3.1. ESTUDIO DE ALTERNATIVAS

A la hora de llevar a cabo la implementación del juego, se plantearon y estudiaron las siguientes cuestiones antes de comenzar a implementar el juego: i) dispositivos para los que desarrollar el juego, ii) lenguaje de programación, iii) juego configurable en local o servidor y iv) diferentes tecnologías del lenguaje *Java*.

A la hora de decidir el dispositivo para el que se desarrollaba el juego se decidió *PC* debido sobre todo a la conveniencia de dotar al juego de contenido multimedia (imágenes), y la visibilidad de éstas en un dispositivo *PDA* estaría comprometida debido al pequeño espacio del que se dispone. Al desarrollarlo para *PC* se pretende contar con un mayor espacio para la interfaz y así poder acompañarlo de contenido multimedia. De esta manera, el juego sería más atractivo para los usuarios y además, se podría incluir tanto fragmentos del plano de metro como iconos de las líneas, ambos lo más parecidos posibles a los reales, para así facilitar el aprendizaje y la generalización al plano real del metro de Madrid. Además un alto porcentaje de las personas con síndrome de Down tienen problemas de visión. Por tanto, necesitamos que las imágenes se muestren a un tamaño adecuado.

Una vez decidido el dispositivo, se eligió como lenguaje de programación *Java*. Además, se barajaron los pros y contras de la implementación del juego en modo local o servidor. Por un lado implementar el juego configurable para modo local tiene la ventaja de dar la posibilidad a los usuarios de jugar fuera del entorno educativo, pero no permite realizar una monitorización de los avances en el juego al no disponer de los resultados del juego de cada usuario y no poder analizar éstos posteriormente. Por esta última razón se decide implementarlo en modo servidor, ya que también se va a desarrollar una aplicación destinada a las personas encargadas del entrenamiento que permite un seguimiento de los resultados obtenidos por cada jugador y en el futuro está previsto ampliarla permitiendo que puedan realizar adaptaciones de presentación o contenidos a usuarios en concreto.

Para finalizar, se planteó la implementación de la parte lógica de la aplicación mediante *applets* o *servlets*. Se descarta la utilización de *applets* ya que las aplicaciones web que los contienen son lentas y es una tecnología que se encuentra en desuso. Sin embargo mediante *servlets* se tiene acceso a bases de datos, a leer y escribir en ficheros, son rápidos, permiten el seguimiento de sesiones, etc. La interfaz de usuario se decidió realizarla utilizando *JSP* ya que se trata al igual que los *servlets* de una tecnología específica para el desarrollo de aplicaciones web y además se cuenta con un conocimiento previo de lenguaje *HTML*. En resumen, nuestra aplicación web realiza tareas de procesado y presentación, siendo los *servlets* adecuados para el procesado y las páginas *JSP* para la presentación. En la tabla 3.1. podemos observar esta repartición de tareas:

SERVLETS	JSP
Procesado de parámetros de la petición	Presentación (vistas)
Acceso a bases de datos	
Lógica de la aplicación	

Tabla 3.1.- Repartición de tareas entre *servlets* y *JSP* en la aplicación

La aplicación desarrollada sigue el **patrón modelo-vista, controlador**. Se trata de un patrón de diseño de aplicaciones *software* donde se separa la lógica de negocio de la interfaz

de usuario [16]. Este modelo divide una aplicación en tres módulos claramente identificables y con funcionalidad bien definida: el modelo, las vistas y el controlador. El modelo representa la información del mundo real que el sistema debe procesar, sin tener en cuenta la forma en la que esa información va a ser mostrada ni los mecanismos que hacen que esos datos estén dentro del modelo. El controlador se encarga de dirigir el flujo de control de la aplicación debido a mensajes externos, como datos introducidos por el usuario u opciones del menú seleccionadas por él. A partir de estos mensajes, el controlador modifica el modelo o abre y cierra vistas. El controlador tiene acceso al modelo y a las vistas, pero las vistas y el modelo no conocen de la existencia del controlador. La vista se encarga de mostrar al usuario la información contenida en el modelo [17].

La utilización de este patrón de diseño tiene una serie de ventajas como son que la aplicación esté implementada modularmente. Esto permite que las aplicaciones que implementan este modelo presenten una extensibilidad y capacidad de mantenimiento mayores en comparación con otras aplicaciones basadas en otros patrones.

En la aplicación desarrollada, el modelo lo compone una base de datos que almacena la información relativa a los jugadores que están registrados en el juego. El controlador está compuesto por una serie de *servlets* que llevan a cabo las acciones pertinentes en función de los datos introducidos por el jugador. Por último, la vista está compuesta por una serie de páginas *JSP* que en función de los datos que le proporcionan los *servlets* muestran cambios en la interfaz.

El flujo de control que se produce en la aplicación es el siguiente (ver Figura 3.1.):

1. El usuario realiza una acción en la interfaz, o lo que es lo mismo se envía una petición *HTTP* al controlador (un *servlet*).
2. El *servlet* procesa la petición de entrada y si es necesario se conecta a la base de datos.
3. Se genera una nueva vista que toma los datos del modelo. El *JSP* lee los parámetros y devuelve la respuesta formateada visualmente al usuario.
4. La interfaz de usuario espera una nueva interacción del usuario, hará que se inicie otro nuevo ciclo.

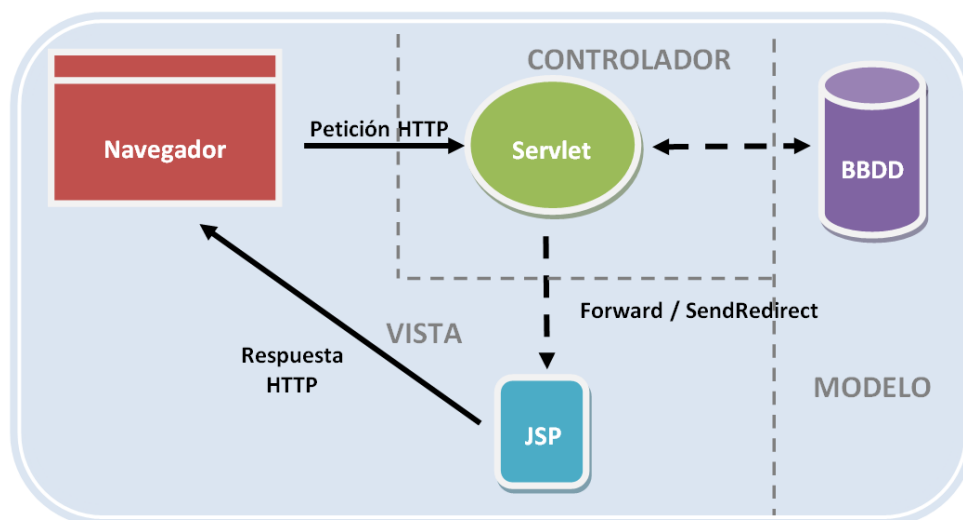


Figura 3.1.- Patrón modelo-vista-controlador

3.2. HERRAMIENTAS Y TECNOLOGÍAS EMPLEADAS

En este apartado se hace una enumeración y descripción de los lenguajes y herramientas utilizados en el diseño e implementación de la aplicación. Son lenguajes orientados a la programación web y las herramientas, en su mayoría gratuitas, se encuentran disponibles para su descarga a través de Internet.

⇒ **HTML (HyperText Markup Language)** [18] [19]

La interfaz gráfica de usuario (GUI) que el juego ofrece al usuario y con la que éste interactúa está implementada en *HTML*. Es el lenguaje de marcado más extendido para el diseño y construcción de páginas web. Se trata de un lenguaje muy sencillo e interpretado (no compilado) que permite crear documentos de hipertexto, que es el formato estándar de las páginas web, los cuáles pueden ser visualizados en múltiples sistemas operativos. La presentación la realizan los navegadores (como *Internet Explorer* o *Mozilla Firefox*). Los documentos *HTML* contienen además de hipertexto enlaces, que conducen a otros documentos, elementos multimedia como imágenes, audio, video, *JavaScript*, etc. Se ha optado por realizar la interfaz gráfica de usuario en *HTML* por ser el lenguaje más extendido en la web, por su sencillez, versatilidad y por el conocimiento previo del mismo.

⇒ **JAVASCRIPT** [19] [20]

JavaScript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de determinadas interactividades con el usuario. Al igual que *HTML*, se trata de un lenguaje de programación interpretado. Todos los navegadores modernos interpretan el código *JavaScript* integrado dentro de las páginas *HTML*. Se ha recurrido a la utilización de *JavaScript* por la necesidad de crear páginas web que incorporen efectos como imágenes que aparecen y desaparecen, acciones que se activan al pulsar un botón, al mover el ratón, etc.

⇒ **CSS (Cascading Style Sheets)** [21]

Las hojas de estilo en cascada permiten dar formato a documentos *HTML*, separando el contenido de la presentación. Son un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre el estilo y formato de múltiples páginas web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la *CSS* afectará a todas las páginas vinculadas a esa *CSS* en las que aparezca ese elemento.

⇒ **Java** [22] [23] [24] [25]

Java es un lenguaje de programación orientado a objetos diseñado por *Sun Microsystems* y perteneciente ahora a Oracle. La compañía *Sun* lo describe como un lenguaje “simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico”. El código se puede escribir en cualquier editor de texto y para compilar el código en *bytecodes*, sólo hace falta descargar gratuitamente la versión del *JDK (Java Development Kit)* deseada.

Para hacer más cómoda la programación en *Java* contamos con los *IDEs (Integrated Development Environment)*, que son entornos de desarrollo integrados. Estos entornos permiten desarrollar las aplicaciones de forma mucha más rápida, incorporando en muchos casos librerías con componentes ya desarrollados, los cuales se añaden al proyecto o programa. Los más extendidos son *Eclipse* y *NetBeans*. Ambos entornos permiten programación de aplicaciones de consola, de aplicaciones web y de aplicaciones visuales.

Java tiene un gran potencial a través de los *servlets Java* y las páginas *JSP* que proporcionan una tecnología segura, sólida e independiente de la plataforma. Por ello la aplicación web desarrollada en este proyecto se ha realizado mediante la combinación de ambas tecnologías.

Los **servlets**, son aplicaciones sin interfaz gráfica que se ejecutan en un servidor de aplicaciones web. Proporcionan un mecanismo para ejecutar programas en equipos servidores, en función de las peticiones que los clientes realicen haciendo uso de navegadores web. El *JSDK (Java Server Developer Kit)*, distribuido gratuitamente por *Sun*, proporciona el conjunto de herramientas necesarias para el desarrollo de *servlets*. Los *servlets* proporcionan las mismas ventajas del lenguaje *Java* en cuanto a portabilidad y seguridad, ya que un *servlet* es una clase de *Java* igual que cualquier otra, y por tanto tiene todas las características del lenguaje. Las características de los *servlets* que los hacen idóneos para este proyecto son:

- Los *servlets* son independientes del servidor utilizado y de su sistema operativo.
- Pueden llamar a otro *servlet*, e incluso a métodos concretos de otros *servlets* distribuyendo así de forma más eficiente el trabajo a realizar.
- Una vez que son llamados por primera vez, quedan activos en la memoria del servidor hasta que el programa que controla el servidor los desactiva. De esta manera se minimiza el tiempo de respuesta, aumentando el rendimiento de la aplicación.
- Permiten la utilización de sesiones, de manera que se puede guardar información específica acerca de un usuario en concreto.
- Se benefician de la gran capacidad de *Java* para ejecutar métodos en ordenadores remotos, conectar con bases de datos, en la seguridad de la información, etc. De hecho una de las tareas más importantes y más frecuentemente realizadas por los *servlets* es la conexión a bases de datos mediante *JDBC (Java DataBase Connectivity)*, API de *Java* que proporciona clases para conectarse con bases de datos.

Aunque los *servlets* pueden ser un programa completo para la generación de respuestas atendiendo a peticiones del cliente, la tarea de generación del contenido dinámico de la interfaz de usuario debe ser separada, para facilitar la programación y reducir en lo posible el coste de creación y mantenimiento. Es entonces cuando se hace necesario incluir la tecnología *JSP* en nuestra aplicación.

JSP (*Java Server Pages*) es una tecnología orientada a crear páginas web que nos permiten separar el *HTML* estático de la parte dinámica implementada con código *Java*. Las páginas *JSP* son una solución que se pueden combinar fácilmente con *servlets*, creando el contenido dinámico, y haciendo más fácil y más rápido construir las aplicaciones basadas en Web que trabajan con una variedad de otras tecnologías: servidores Web, navegadores Web, servidores de aplicación y otras herramientas de desarrollo. En realidad, las páginas *JSP* nos permiten poner fragmentos de código directamente dentro de una página *HTML* estática. Cuando el navegador carga una página *JSP*, se ejecuta el código del *servlet* y el servidor de aplicaciones crea, compila, carga y ejecuta un *servlet* en segundo plano para ejecutar los segmentos de código *servlet* y devolver una página *HTML*. Con *JSP* podemos crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que *Java* es en esencia un lenguaje multiplataforma.

⇒ **SGBD MySQL** [26]

Para el manejo y creación del almacén de datos de los jugadores se ha empleado el Sistema Gestor de Bases de Datos (SGBD) *MySQL*. Proporciona un servidor de base de datos relacional *SQL* muy rápido, multihilo y multiusuario. Contiene una completísima documentación y gran cantidad de ayuda disponible en la web.

MySQL es actualmente uno de los SGBD más extendidos y utilizados. Su éxito reside en que no depende de la plataforma, existiendo versiones para distintos sistemas operativos. Otra de las ventajas que posee es que es gratuito (para usos con licencia GNU, si bien las empresas que deseen la incorporación en productos privativos deben comprar una licencia que les permita ese uso) y soporta múltiples accesos simultáneos (aunque sin llegar a los grandes volúmenes soportados por bases de datos como Oracle). Para trabajar con el lenguaje *Java* incluye el Connector/J, driver oficial *JDBC*, para interconectarlos. El sistema operativo del servidor donde estará alojada la aplicación es *GNU/Linux*. Como consecuencia de esto, se descarta por completo el uso de *MS SQL Server* y *Access*. Además, se busca que las tecnologías con las que se trabajan sean *software* libre. En este ámbito Oracle por ser un sistema de pago y muy costoso queda descartado.

⇒ **GIMP 2.6.0 (GNU Image Manipulation Program)** [27]

GIMP es un programa de tratamiento de imágenes libre. Es ideal para tareas como retocar fotografías, crear imágenes para la web, etc. Se distribuye bajo una licencia libre, y puede usarse tanto en sistemas *Unix* como *Windows*. Los motivos de elección de este programa para realizar estas tareas han sido varios: su licencia *GNU*, porque reconoce el formato de imagen *.png* (formato en el que se encuentran las imágenes utilizadas en la *GUI* del juego) y por su facilidad de uso. Este programa se ha utilizado en este proyecto para la edición de imágenes (encabezados, botones, planos de metro), en concreto para escalar, recortar y ajustar las imágenes utilizadas.

⇒ **Microsoft Office PowerPoint 2007** [28]

Este conocido programa desarrollado por Microsoft e incluido en el paquete Office permite crear rápidamente presentaciones atractivas y dinámicas de una forma sencilla e intuitiva. Este programa se ha utilizado con dos objetivos fundamentales:

- Dada su facilidad de uso y la posibilidad de poder realizar diagramas, insertar y dar formato a imágenes, etc. se ha usado para crear un primer prototipo del diseño de la interfaz de usuario para presentársela a los responsables del proyecto destinatario.
- Se ha utilizado para crear la mayoría de las imágenes que aparecen en la interfaz de usuario ya que ofrece una gran variedad de trabajo con imágenes prediseñadas en múltiples formatos (3D, sombras, colores, biselados...) y la posibilidad de una vez editadas guardarlas en formato *.png*. Algunas de ellas se presentan en la Figura 3.2.

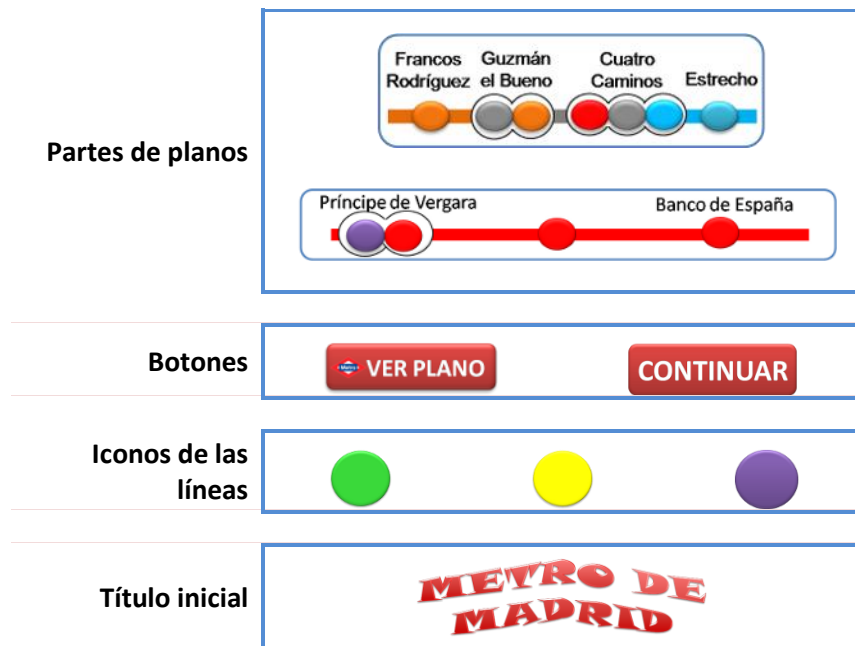


Figura 3.2.- Imágenes diseñadas con Microsoft Office Power Point 2007.

⇒ **Plataforma IDE NetBeans 6.7.1.** [29]

NetBeans 6.7.1. es un entorno de desarrollo integrado de software de código abierto, libre. Fue creado por *Sun Microsystems* y está implementado en *Java* aunque se pueden desarrollar aplicaciones en múltiples lenguajes de programación como *Java*, *C*, *C++*, *Ruby*, *Python* y *PHP*. Al ser un *software* de código abierto, existen multitud de módulos que extienden su funcionalidad. Contiene prácticamente todo lo que se le suele pedir a un IDE, editor avanzado de código, depurador, diversos lenguajes y extensiones de todo tipo.

Sus principales ventajas a la hora de afrontar este proyecto radican en la multitud de herramientas que posee el entorno en la instalación básica, que facilitan de manera sustancial tareas propias de programación web, además de incluir los servidores de aplicaciones *GlassFish* y *Tomcat*. En concreto las características más importantes de este entorno son, la disponibilidad de la integración de sintaxis, de código y de comprobación de errores de código *JavaScript* junto con la capacidad de edición de las *CSS* permitiendo previsualizar los cambios. La única desventaja es que es un entorno que consume muchos recursos computacionales.

⇒ **Servidor de aplicaciones GlassFish** [30]

GlassFish es un servidor de aplicaciones de código abierto desarrollado por *Sun Microsystems* que implementa *Java EE 5*. Está basado en el servidor *Sun Java System Enterprise Server*, un derivado de *Apache Tomcat*. Soporta las versiones de tecnologías más recientes incluidas en *J2EE 5*, como por ejemplo *JavaServer Pages 2.1*, *JavaServerFaces 1.2*, *Servlet 2.5*, *Enterprise JavaBeans 3.0* y *Java API for Web Services 2.0*. Aprovechamos la integración de este servidor en el IDE *NetBeans 6.7.1*, para hacer más liviano el proceso de depuración y detección de errores de la aplicación.

3.3. ORGANIZACIÓN Y ESTRUCTURA DE CARPETAS Y FICHEROS

La aplicación web completa en el servidor se encuentra en un directorio que contiene la siguiente estructura de subdirectorios:

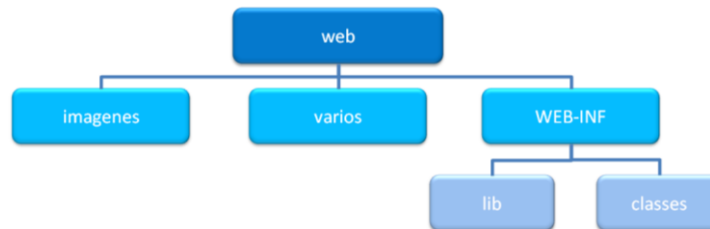


Figura 3.3.- Organización de los principales directorios de la aplicación

El directorio raíz (**web**) contiene los ficheros *JSP* que conforman la interfaz de usuario del juego. La Tabla 3.2. presenta cada uno de los ficheros *JSP* junto con su contenido asociado.

NOMBRE	CONTENIDO
<i>paginaprincipal.jsp</i>	Pantalla de comienzo del juego.
<i>elegir_opcion.jsp</i>	Formulario para que el jugador seleccione la opción “nuevo jugador”, si no está registrado en el juego, o “continuar partida” si ya está registrado y quiere seguir jugando.
<i>registro_usuario.jsp</i>	Formulario de registro de un nuevo usuario. Se solicita al jugador que escriba el nombre con el que quiere participar en el juego.
<i>registro_error.jsp</i>	Muestra un mensaje de error al jugador si se ha introducido en el registro de nuevo usuario un nombre existente en el sistema. Vuelve a mostrar el formulario de registro para que elija un nuevo nombre.
<i>registro_correcto.jsp</i>	Muestra la confirmación de registro correcto de un nuevo jugador.
<i>continuar_partida.jsp</i>	Formulario de validación de un jugador en el sistema. El usuario tiene que escribir el nombre con el que se registró en el juego.
<i>usuario_error.jsp</i>	Mensaje de error en la validación de un jugador en el sistema si el nombre introducido no se corresponde con ningún usuario registrado. Se vuelve a mostrar el formulario de validación.
<i>usuario_correcto.jsp</i>	Muestra la confirmación de validación correcta de un jugador en el juego si el nombre introducido en la pantalla <i>continuar_partida.jsp</i> se encuentra en la base de datos de la aplicación.
<i>nivel_resumen.jsp</i>	Muestra las puntuaciones obtenidas en cada nivel y los botones correspondientes para jugar una ronda de los niveles a los que tiene

	acceso el jugador.
<i>nivel_comienzo.jsp</i>	Muestra el nivel seleccionado para jugar una ronda y redirecciona a la primera pregunta de la ronda.
<i>nivelX_pregunta.jsp</i>	Muestra los enunciados de las preguntas del <i>nivel X</i> y las respuestas posibles. X son los niveles en los que se estructura la aplicación y que van de 1 a 4.
<i>nivel_similitud.jsp</i>	Formulario para que el jugador decida si quiere sustituir la respuesta que ha escrito en la caja de texto en una pregunta del tipo 1 del <i>nivel 4</i> por la corrección que le ofrece el juego.
<i>nivel_respuesta.jsp</i>	Muestra la comparación entre la respuesta del jugador a la pregunta planteada y la respuesta correcta junto con un mensaje de ánimo.
<i>nivel_fin.jsp</i>	Pantalla de fin de ronda. Muestra la puntuación obtenida para ese nivel junto con un mensaje de ánimo.
<i>plano_metro.jsp</i>	Muestra el plano del metro de Madrid.
<i>salir.jsp</i>	Formulario para que el jugador confirme que desea salir del juego.
<i>error.jsp</i>	Pantalla que muestra un mensaje de error y cierra el juego.

Tabla 3.2.- Ficheros JSP del directorio web.

Además, dentro de este directorio hay 3 subdirectorios: imagenes, varios y WEB-INF. En el directorio **imagenes**, se encuentran organizados todos los elementos gráficos que se pueden ver en las distintas pantallas de la aplicación (botones, mapas, iconos, medallas, etc.). El directorio **varios** contiene distintos ficheros que necesita la aplicación para su funcionamiento. A continuación se describe el contenido de este directorio.

- **Ficheros de texto:** Hay 4 ficheros de texto con los distintos mensajes de ánimo que se muestran en la pantalla de respuesta y 13 ficheros de texto con los nombres de las estaciones de metro, en concreto hay un fichero por cada línea de metro. Además hay un fichero que contiene tres líneas que se corresponden respectivamente con el usuario y la contraseña del servidor de *MySQL* y con el servidor de la aplicación (ver Tabla 3.3.).

NOMBRE	CONTENIDO
<i>acierto1.txt</i>	6 frases que le indican al jugador que la respuesta proporcionada es correcta.
<i>acierto2.txt</i>	6 frases de ánimo para continuar con el juego en caso de que la respuesta dada por el jugador sea correcta.
<i>error1.txt</i>	6 frases que le indican al jugador que la respuesta proporcionada es incorrecta.
<i>error2.txt</i>	6 frases de ánimo para continuar con el juego en caso de que la respuesta dada por el jugador sea incorrecta.
<i>lineaX.txt</i>	Nombre de las estaciones de la línea X. X son las 12 líneas de metro de Madrid junto con el ramal numerada como 13 para la aplicación.
<i>configuracion acceso bd.txt</i>	Usuario y contraseña de acceso al servidor de <i>MySQL</i> , así como servidor de la aplicación.

Tabla 3.3.- Ficheros de texto del directorio varios.

- **Fichero de estilo (*estilos.css*)** utilizado para formatear la interfaz del programa.

- *Fichero de javascript (utilidades.js)* contiene las funciones que son referenciadas desde las páginas JSP. En la siguiente tabla se muestran las funciones javascript implementadas en el fichero.

NOMBRE	FUNCIONALIDAD
<i>Respuesta(seccion1, seccion2, seccion3, seccion4)</i>	En las preguntas con 2 respuestas posibles, donde sólo una es la correcta, muestra, en la zona de la respuesta seleccionada, la última respuesta elegida por el jugador y la oculta en la zona de respuestas posibles. Asigna como respuesta el valor de la respuesta seleccionada por el jugador en cada momento.
<i>Respuesta2(seccion1, seccion2, seccion3, seccion4, seccion5, seccion6)</i>	En las preguntas con 3 respuestas posibles, tanto en las que la respuesta correcta está formada por una como por dos de las respuestas posibles, muestra en la zona de la respuesta seleccionada la última o dos últimas respuestas elegidas, según el caso, por el jugador y las oculta en la zona de respuestas posibles. Asigna como respuestas los valores de las respuestas seleccionadas por el jugador en cada momento.
<i>Respuesta3(seccion1, seccion2, seccion3, seccion4, seccion5, seccion6, seccion7, seccion8)</i>	En las preguntas con 4 respuestas posibles, donde la respuesta correcta se compone por 3 de las 4 respuestas posibles, muestra en la zona de la respuesta seleccionada las 3 últimas respuestas elegidas por el jugador y las oculta en la zona de respuestas posibles. Asigna como respuestas los valores de las respuestas seleccionadas por el jugador en cada momento.
<i>MostrarPlano()</i>	Abre una ventana de 600x100 sin barra de herramientas y sin menú con el contenido del plano del metro cuando el jugador hace clic en el botón "Ver plano".
<i>conMayusculas(field)</i>	Convierte el valor recibido a mayúsculas. Se utiliza cuando el jugador introduce el nombre de usuario en una caja de texto tanto para registrarse como para validarse en el juego.
<i>ReemplazarCaracteres()</i>	Reemplaza en el valor de las respuestas dadas por el jugador los caracteres ASCII á, é, í, ó, ú, Á, É, Í, Ó, Ú, ñ y ü por su equivalente en código HTML. Se utiliza en las preguntas en las que las respuestas son nombres de estaciones.
<i>ObtenerRespuestas(tipo)</i>	Asigna el contenido que el jugador ha escrito en las cajas de texto a ciertos elementos del documento para luego ser recuperados desde el <i>servlet</i> correspondiente. Además utiliza la función <i>ReemplazarCaracteres()</i> .
<i>CambiarRespuesta()</i>	En las preguntas del tipo 1 del <i>nivel 4</i> si el jugador acepta la corrección que se le propone a la respuesta que él ha escrito, se sustituye la respuesta.
<i>NoCambiarRespuesta()</i>	En las preguntas del tipo 1 del <i>nivel 4</i> si el jugador no desea realizar la corrección que le propone el juego mantiene la respuesta que había escrito éste.
<i>PermitirSoloNumeros(evt)</i>	Sólo permite introducir números por teclado en las cajas de texto de las preguntas de los tipos 5 y 6 del <i>nivel 4</i> .

Tabla 3.4.- Funciones Javascript implementadas en el fichero *utilidades.js* del directorio *varios*

- **Ficheros JSP** cuyo contenido se incluye en los ficheros *JSP* mediante la directiva *include* (`<jsp:include page="url"/>`). Hay elementos que están en distintas paginas *JSP* (ver Tabla 3.5.), como por ejemplo la barra de progreso de puntuaciones que se muestra entre otros en *nivel1_pregunta.jsp*, *nivel2_pregunta.jsp*, *nivel_respuesta.jsp*, etc. Por ello hay fragmentos de código que se han extraído a ficheros *JSP* independientes, evitando así la redundancia de código.

NOMBRE	FUNCIONALIDAD
<i>boton_salir.jsp</i>	Muestra el botón de salir de la aplicación.
<i>encabezado.jsp</i>	Muestra el encabezado que corresponda en cada momento en las páginas de pregunta, respuesta y de fin de nivel.
<i>enunciado_pregunta.jsp</i>	Muestra el enunciado de la pregunta actual. En las preguntas que corresponde, también muestra la imagen que acompaña al enunciado.
<i>puntuacion.jsp</i>	Muestra la puntuación del nivel que se está jugando mediante una barra de progreso en las páginas de pregunta y respuesta.
<i>recordatorio_respuesta.jsp</i>	En el caso que el jugador no haya seleccionado el número de respuestas necesarias para dar respuesta a la pregunta actual muestra un mensaje recordándole ese número.

Tabla 3.5.- Ficheros JSP del directorio varios

El directorio **WEB-INF** contiene el fichero *web.xml*. Este fichero configura la aplicación. Es donde se han declarado los servlets, si hubiera sido necesario es también donde se les habrían asignado los parámetros de inicio, declarado los alias y los filtros. Además del fichero *web.xml* tiene 2 directorios. El directorio *classes* y el directorio *lib*.

- El directorio **classes** es en el que están los ficheros compilados de las clases (incluye los *servlets*) de la aplicación web.

NOMBRE	FUNCIONALIDAD
<i>Vertice.class</i>	Define la clase <i>Vertice</i> , sus atributos y métodos. Cada objeto de esta clase se corresponde con una estación de la red de metro de Madrid.
<i>GrafMatPeso.class</i>	Define la clase <i>GrafMatPeso</i> , sus atributos y métodos. Un objeto de esta clase es la red de metro de Madrid.
<i>TodoCaminoMinimo.class</i>	Define la clase <i>TodoCaminoMinimo</i> , sus atributos y métodos. En esta clase se implementan distintos métodos que se necesitan para hallar la respuesta correcta a algunas de las preguntas que se plantean al jugador durante el juego. En concreto aquellas preguntas que hacen preguntas acerca del número de estaciones que componen un trayecto, el nombre de las estaciones en las que hay que hacer transbordo en un recorrido, etc.
<i>AuxPreguntas.class</i>	Esta clase tiene una serie de métodos auxiliares que utilizan los 4 servlets de pregunta y el de respuesta.
<i>servlet_altausuario.class</i>	Valida que el nombre escrito por un usuario en el formulario de registro de un nuevo jugador no existe en el sistema. Si no existe se le da de alta en la base de datos y si existe se informa al usuario.

<i>servlet_existeusuario.class</i>	Realiza la validación de un usuario en el juego. Si no existe el nombre introducido pide uno nuevo. En caso de una validación correcta recupera las puntuaciones del jugador almacenadas en la base de datos para cada nivel.
<i>servlet_inicializarnivel.class</i>	Obtiene los puntos obtenidos para cada tipo de pregunta del nivel del que ha seleccionado jugar una ronda e inicializa otros parámetros (número de pregunta, nivel, etc.) antes de comenzar la ronda.
<i>servlet_preguntaX.class</i>	Elige el tipo de pregunta del <i>nivel X</i> del que el jugador tiene menos puntos y selecciona una pregunta de ese tipo junto con las respuestas posibles. Los niveles van de 1 a 4 como ya se ha explicado anteriormente ($x \geq 1$ y $x \leq 4$).
<i>servlet_respuesta.class</i>	Compara la/s respuesta/s dada/s por el usuario a la pregunta planteada con las respuesta/s correcta/s.
<i>servlet_finalizarnivel.class</i>	El juego actualiza las puntuaciones del nivel jugado en la base de datos al finalizar una ronda. En caso de haber obtenido una puntuación inferior a 80 en un nivel ya superado pone a 0 los resultados de los niveles superiores.
<i>servlet_finjuego.class</i>	Invalida la sesión del jugador que sale del juego.

Tabla 3.6.- Ficheros servlet del directorio classes

- Directorio *lib* contiene la biblioteca (comprimida con jar) *mysql-connector-java-5.1.12-bin* que utiliza la aplicación.

3.4. BASE DE DATOS

En este punto se va a analizar la base de datos que se ha utilizado en el proyecto. Se pretende recoger la estructura de la base de datos indicando el nombre de la tabla, el nombre de los diferentes campos que la componen y tipos de datos de los diferentes campos.

La base de datos “jugadores” está implementada en *MySQL* y está compuesta únicamente por la tabla “*datosjugadores*”. Esta tabla contiene, como su nombre indica, la información de los usuarios registrados en el juego. Almacena para cada jugador su nombre, el número de preguntas que ha contestado de cada nivel y el número de preguntas acertadas de cada tipo. Está compuesta por los siguientes campos que almacenan la información que se expone en la Tabla 3.7.

CAMPO	DEFINICIÓN
nombre	Es la clave principal. Contiene el nombre de un jugador registrado.
preguntas1	Número de preguntas del <i>nivel 1</i> que ha contestado el jugador.
aciertos11	Número de preguntas del tipo 1 y <i>nivel 1</i> acertadas por el jugador.
...	
aciertos16	Número de preguntas del tipo 6 y <i>nivel 1</i> acertadas por el jugador.
preguntas2	Número de preguntas del <i>nivel 2</i> que ha contestado el jugador.
aciertos21	Número de preguntas del tipo 1 y <i>nivel 2</i> acertadas por el jugador.
...	
aciertos26	Número de preguntas del tipo 6 y <i>nivel 2</i> acertadas por el jugador.

preguntas3	Número de preguntas del <i>nivel 3</i> que ha contestado el jugador.
aciertos31	Número de preguntas del tipo 1 y <i>nivel 3</i> acertadas por el jugador.
...	
aciertos36	Número de preguntas del tipo 6 y <i>nivel 3</i> acertadas por el jugador
preguntas4	Número de preguntas del <i>nivel 4</i> que ha contestado el jugador.
aciertos41	Número de preguntas del tipo 1 y <i>nivel 4</i> acertadas por el jugador
...	
aciertos46	Número de preguntas del tipo 6 y <i>nivel 4</i> acertadas por el jugador

Tabla 3.7.- Campos de la tabla "datosjugadores"

Cuando se da de alta a un nuevo jugador se almacena en el campo *nombre* el contenido que ha escrito en la caja de texto el usuario y el resto de campos se inicializan con el valor por defecto '0'. El nombre de los jugadores se almacena en mayúsculas y sin tildes. Cuando un jugador se valida en el juego también se transforma el contenido que escribe a mayúsculas y sin tildes para realizar la comparación. La actualización de los valores de las puntuaciones sólo se realiza cuando el jugador finaliza cada ronda.

3.5. IMPLEMENTACIÓN DE CLASES

A la hora de implementar el juego lo primero es hallar una representación del plano del metro de Madrid que permita realizar las operaciones que necesita el juego para su funcionamiento (p.e. calcular el número de estaciones de un trayecto, hallar las estaciones en las que hay que realizar transbordo, etc.). Ya a primera vista se equipara a la estructura de un grafo, definido como un conjunto de vértices, tantos como estaciones existen, unidos entre sí mediante selecciones de pares estación-estación a través de aristas, que representan tramos de línea entre dos estaciones.

Un grafo $G=(V,E)$ es una pareja ordenada de conjuntos de vértices y arcos. Generalmente se representan con un dibujo donde cada vértice es un punto y cada arista es una línea que une dos puntos. Los grafos son estructuras con las que se formulan muchos problemas por medio de objetos y las conexiones que hay entre ellos, como son los mapas de carreteras, sistemas de telecomunicaciones, representación de circuitos electrónicos, redes de ordenadores o redes de transporte (ver Figura 3.4.).

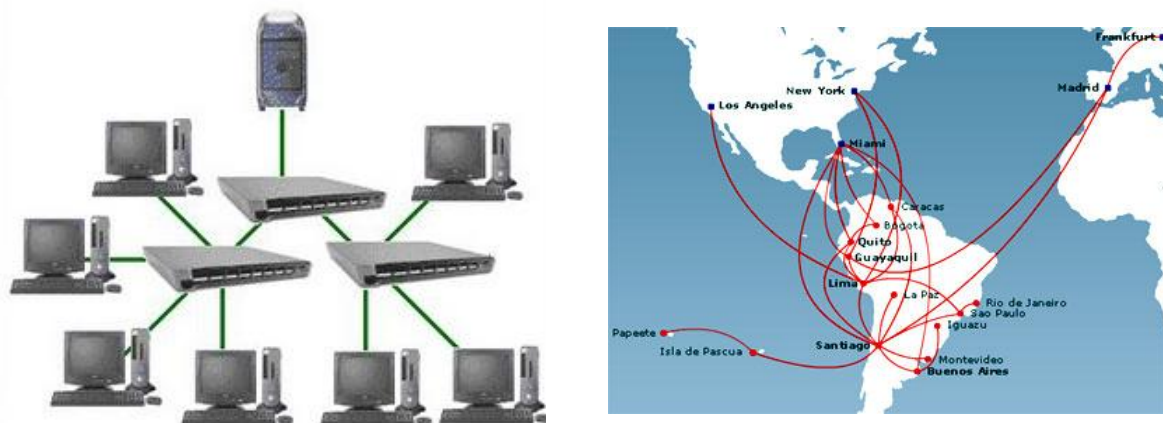


Figura 3.4.- Distintos sistemas reales con estructura de grafo

Por ello se ha representado la red de metro de Madrid mediante la implementación de un grafo. En concreto, se trata de un grafo simple, puesto que sólo una arista (tramo de línea) cualquiera une dos estaciones concretas; es un grafo no dirigido ya que se puede realizar el recorrido de una estación a otra en ambos sentidos; conexo, porque toda estación está conectada al menos con otra a través de un tramo de línea, pero incompleto, porque no hay aristas para todos los posibles pares de estaciones, esto es, no todas las estaciones se encuentran unidas entre sí directamente mediante aristas. Si bien un grafo es una representación válida de una red de metro, se ha tenido en cuenta a la hora de realizar la implementación de las clases que el metro de Madrid es una estructura compuesta por líneas, que están formadas por estaciones y éstas a su vez pueden pertenecer a más de una línea, debido a las intersecciones entre líneas.

A la hora de elegir la forma de implementar la estructura grafo para el metro de Madrid se ha optado por hacerlo mediante una matriz de adyacencia ponderada. Esta decisión se ha tomado porque la operación principal que realiza la aplicación sobre la estructura grafo es obtener la longitud de caminos mínimos entre pares de nodos y como se comenta más adelante para realizar estos cálculos se ha implementado el algoritmo de Floyd que trabaja con matrices de adyacencia con peso.

En total para el desarrollo del juego se han implementado 4 clases: **Vertice**, **GrafMatPeso**, **CaminosMinimos** y **AuxPreguntas**, de las que se hace una descripción a continuación.

En la aplicación cada objeto de la clase *Vertice* representa a una estación de la red de metro de Madrid. La clase *Vertice* implementa la información que se almacena en cada vértice del grafo. Los atributos son el nombre de la estación, el conjunto de líneas a las que pertenece la estación y el número de vértice de la estación en el grafo.

La clase *Vertice* tiene dos métodos constructores, en uno de ellos se le indica el nombre de la estación y la línea a la que pertenece y en el otro tan sólo el nombre de la estación. Hay además, un método que añade al conjunto de líneas a las que pertenece la estación una nueva línea dada. Un tercer método devuelve el atributo que indica el nombre de la estación. Otro método realiza la comparación entre dos vértices y un último método establece el número de orden del vértice dentro del grafo.

La clase *GrafMatPeso* representa un objeto de tipo grafo para poder almacenar la información del mapa de metro de Madrid. Cada uno de los objetos de la clase *GrafMatPeso* tendrá 9 atributos y 8 métodos (ver Figura 3.5.). Los atributos son: el valor que contiene la matriz del grafo en la intersección de las estaciones que no tienen conexión entre ellas (dos vértices sin un arco entre ellos están a distancia INFINITO), una matriz bidimensional de enteros que representa el grafo, el número de nodos del grafo, la capacidad máxima del grafo, un *array* que almacena todos los objetos *Vertice* del grafo y 4 atributos donde se almacenan las estaciones que pertenecen a una, dos, tres o cuatro líneas de metro respectivamente.

GrafMatPeso
+ int INFINITO = 0xFFFF
+ int MatPeso [][]
- int NumVerts
+ int maxVerts
- Vertice vertices []
+ <<vector>> estaciones1
+ <<vector>> estaciones2
+ <<vector>> estaciones3
+ <<vector>> estaciones4

```

+ GrafMatPeso(int max)
+ GrafMatPeso()
+ int NumeroDeVertices()
+ Vertice Vertices()
+ void nuevoVertice(String nom, String l)
+ void nuevoArco(String a, String b, int peso)
+ int numVertice(String nom)
+ void entradaGrafo(String path)

```

Figura 3.5.- Clase *GrafMatPeso*

Respecto a los métodos de la clase *GrafMatPeso* posee dos constructores, un método que devuelve el número de vértices que tiene el grafo (el valor de *numVerts*), otro que devuelve un array que contiene todos los objetos *Vertice* del grafo (el valor de *vertices*), un método que crea y añade un nuevo vértice al grafo, en caso que no exista, si ya existiese realiza las operaciones de actualización necesarias para reflejar la pertenencia del vértice a la línea que se indica (p.e. añade la línea de metro a la relación de líneas a las que pertenece la estación, sitúa la estación en la relación de estaciones correspondiente según el número de líneas a las que pertenezca), un método que almacena en la intersección en *matPeso* de dos estaciones el valor de *peso* (valor 1 en caso que dos estaciones se encuentren conectadas por un tramo de la línea de metro), otro que dado el nombre de una estación devuelve el número de nodo que tiene dentro de la red de metro; y en último lugar una función que lee los 13 ficheros de texto (uno por cada línea de la red de metro de Madrid) que contienen el nombre de las estaciones añadiendo al grafo los correspondientes vértices y aristas.

Uno de los problemas más importantes en los grafos es encontrar los caminos de coste mínimo entre los pares de nodos. En la aplicación ha hecho falta calcular la longitud (entendida como número de aristas) del camino más corto entre pares de estaciones, para ello se ha implementado la clase *CaminosMinimos* con los atributos y métodos necesarios.

Para calcular el menor de los caminos mínimos entre dos estaciones cualesquiera del grafo se parte, como ya se ha explicado anteriormente, que el grafo que forma el metro de Madrid es conexo, por lo que siempre habrá un camino entre todo par de estaciones. La cuestión es encontrar el camino de menor longitud entre dos estaciones dadas. Para ello se ha aplicado el algoritmo de Floyd (ver Figura 3.6.) que, dada una matriz de adyacencia *M* del grafo *g*, calcula una matriz *D* que guarda los resultados del algoritmo de Floyd (longitud del camino mínimo que une cada par de vértices), y otra matriz que guarda la sucesión de nodos que componen los caminos mínimos llamada *traza*.

```

procedure FLOYD (M)
begin
   $D^{(0)} = M;$ 
  for k:=1 to n do
    for i:=1 to n do
      for j:=1 to n do {
         $d_{i,j}^{(k)} := \min(d_{i,j}^{(k-1)}, d_{i,k}^{(k-1)} + d_{k,j}^{(k-1)});$ 
        traza[i][j]=k;
      }

```

Figura 3.6.- Algoritmo de Floyd

Ha sido necesario modificar el algoritmo para obtener los caminos mínimos de forma que se adapte a nuestras necesidades. En el *nivel 1* se ha adaptado de manera que calcula los caminos mínimos entre las estaciones que pertenecen a la línea de metro que se le indica. En el *nivel 2* se calculan los caminos mínimos entre las estaciones que están separadas por un solo transbordo. En el *nivel 3* se ha adaptado para calcular los caminos mínimos entre estaciones que están separadas por dos transbordos. Por último, en el *nivel 4* se utiliza el algoritmo de Floyd propiamente dicho.

La clase *CaminosMinimos* (ver Figura 3.7.) tiene los siguientes atributos: un *array* que contiene los valores de la matriz de adyacencia del grafo, un segundo array que almacena los vértices por los que pasa el trayecto que se está calculando y un último array que contiene el número de vértices que recorre el trayecto que se está calculando. Además, contiene información sobre el número de vértices del grafo, un atributo que en caso de haber más de un transbordo a lo largo de un trayecto adopta el valor *false*, un atributo que almacena las estaciones que componen un trayecto de metro (desde el origen hasta el destino) y por último un atributo que contiene la relación de estaciones en que se hace transbordo en un camino.

CaminosMinimos
+ int pesos [] [] + int traza [] [] + int d [] [] - int n + boolean novale=false + <<vector>> estaciones camino + <<vector>> estaciones transbordo
+ CaminosMinimos(GrafMatPeso gp) + void ObtenerCaminosMinimosSinTransbordo(GrafMatPesto gp, String linea) + void CaminoMinimoUnTransbordo(int i, int j, GrafMatPeso gp, String linea) + void CalcularCaminosMinimos() + void ObtenerRecorrido(int i, int j, GrafMatPeso gp, String linea, int nivelactual) + void ObtenerRecorridoAuxiliar(int i, int j, GrafMatPeso gp)

Figura 3.7.- Clase *CaminosMinimos*

Respecto a los métodos de la clase *CaminosMinimos*, posee un constructor, tres métodos que calculan caminos mínimos: i) entre las estaciones de la línea que se le pasa como parámetro, ii) entre dos estaciones dadas realizando sólo un transbordo, y iii) el algoritmo de Floyd propiamente dicho que calcula todos los caminos mínimos entre todos los pares de vértices del grafo. Por último, una vez que se ha obtenido el camino mínimo deseado, hay un método que obtiene la relación de estaciones que componen el trayecto y en las que se hace transbordo.

La clase *AuxPreguntas* es utilizada de manera auxiliar por varios *servlets*. Se ha creado para evitar tener código redundante, ya que hay operaciones que realizan varios *servlets*. Estas operaciones se han situado en esta clase para que puedan ser usadas por todos ellos. En concreto los *servlets* que hacen uso de esta clase con: *servlet_pregunta1*, *servlet_pregunta2*, *servlet_pregunta3*, *servlet_pregunta4* y *servlet_respuesta*.

Esta clase se compone de varios métodos. El primer método devuelve un número aleatorio comprendido entre el rango de números que se le indica (se utiliza por ejemplo para seleccionar aleatoriamente un número de línea, una estación del grafo, etc.). Existe otro método que devuelve el tipo de pregunta del nivel actual del que el jugador tiene menos aciertos, y por tanto del que tiene que ser la siguiente pregunta planteada. El siguiente método devuelve un número que identifica a una estación de la línea de metro que se le indica. Otro de los métodos se utiliza para dar mayor variedad a las preguntas del tipo 6 de

todos los niveles, para ello devuelve un enunciado seleccionado aleatoriamente de entre una de las plantillas de pregunta que tiene. Hay dos métodos para tratar cadenas de caracteres: uno para sustituir los códigos HTML correspondientes por los caracteres ASCII á, é, í, ó, ú, ñ, ü, Á, É, Í, Ó, Ú, Ñ y Ü y otro para devolver una cadena de caracteres sin tildes y en mayúsculas. Por último, existe un método que se utiliza en el nivel para las preguntas del tipo 1, para comprobar si la respuesta que ha escrito por teclado el jugador difiere en 3 o menos caracteres o si falta un carácter respecto a la respuesta correcta.

3.6 APLICACIÓN PARA REALIZAR SEGUIMIENTO DE RESULTADOS

Como herramienta de apoyo al profesorado se ha implementado una aplicación que permite realizar un seguimiento de los resultados que obtienen los distintos jugadores en el juego. Así, de manera sencilla se puede consultar la puntuación que tiene cada jugador en cada nivel del juego y los tipos de preguntas de cada nivel en los que ha obtenido peores resultados. Para consultar los datos de un determinado usuario, el profesor solamente tendrá que seleccionar el nombre del alumno sobre el que quiere ver su progreso y a continuación se le mostrará una pantalla similar a la que aparece en la Figura 3.8.



	Puntos	Tipo con menos aciertos
NIVEL 1	100	2 - 3 - 4
NIVEL 2	90	2 - 3 - 4
NIVEL 3	90	1 - 2 - 3 - 4 - 5
NIVEL 4	--	--

Figura 3.8 Ejemplo de pantalla de la aplicación “Metro de Madrid – Seguimiento de Resultados”

Mediante el uso de esta aplicación el profesorado podrá monitorizar los resultados personales de cada jugador y ofrecer un apoyo personalizado en función de los tipos de preguntas en los que cada uno encuentre más dificultades.

Además se ha incluido la posibilidad que los profesores puedan realizar una gestión de los jugadores que hay registrados en el juego, en concreto podrán eliminar usuarios. De esta forma podrán borrar jugadores registrados por error.

⇒ Organización y estructura de carpetas y ficheros

Esta aplicación web completa en el servidor se encuentra en un directorio con la misma estructura de subdirectorios que el juego “Metro de Madrid” (ver Figura 3.3.). Además hace uso de los datos almacenados en la base de datos del juego (ver apartado 3.4 - Base de Datos).

El directorio raíz (**web**) contiene los ficheros *JSP* que conforman la interfaz de usuario del juego. La Tabla 3.8. presenta cada uno de los ficheros *JSP* junto con su contenido asociado.

NOMBRE	CONTENIDO
<i>paginaprincipal.jsp</i>	Pantalla de comienzo del juego.
<i>selecciona_jugador.jsp</i>	Formulario con una lista desplegable que contiene los nombres de todos los jugadores registrados en el juego para que el profesor seleccione el nombre que desea y pulse el botón que se corresponda con la acción que quiere realizar con ese jugador (darlo de baja del juego o consultar su progreso en el mismo). Si no hay ningún jugador registrado en el juego muestra un mensaje informando de ello.
<i>resultados_jugador.jsp</i>	Muestra una tabla con información de los resultados del jugador seleccionado en la pantalla anterior. En concreto se presenta la puntuación obtenida en cada nivel y los tipos de preguntas de cada nivel en los que el jugador ha obtenido los peores resultados.
<i>eliminacion_jugador.jsp</i>	Confirma la eliminación del jugador seleccionado en la pantalla previa.

Tabla 3.8.- Ficheros JSP del directorio web.

Además, dentro de este directorio hay 3 subdirectorios: imagenes, varios y WEB-INF. En el directorio **imagenes**, se encuentran organizados todos los elementos gráficos que se pueden ver en las distintas pantallas de la aplicación (botones, fondo de pantalla, etc.). El directorio **varios** contiene distintos ficheros que necesita la aplicación para su funcionamiento. A continuación se describe el contenido de este directorio.

- **Fichero de texto:** Hay un fichero que contiene tres líneas que se corresponden respectivamente con el usuario y la contraseña del servidor de *MySQL* y con el servidor de la aplicación (ver Tabla 3.9.).

NOMBRE	CONTENIDO
<i>configuracion acceso bd.txt</i>	Usuario y contraseña de acceso al servidor de <i>MySQL</i> , así como servidor de la aplicación.

Tabla 3.9.- Fichero de texto del directorio varios.

- **Fichero de estilo** (*estilos.css*) utilizado para formatear la interfaz del programa.
- **Fichero de javascript** (*utilidades.js*) contiene la función que es referenciada desde las páginas *JSP*. En la siguiente tabla se muestra la función *javascript* implementada en el fichero.

NOMBRE	FUNCIONALIDAD
<i>ObtenerNombreyOpcion()</i>	Obtiene el nombre del jugador que se ha seleccionado en la lista desplegable que muestra <i>selecciona_jugador.jsp</i> y la acción que ha seleccionado el usuario realizar con los datos correspondientes al jugador seleccionado.

Tabla 3.10.- Función Javascript implementada en el fichero *utilidades.js* del directorio varios

- Fichero JSP, llamado *boton_salir.jsp*, cuyo contenido se incluye en los ficheros *JSP* mediante la directiva *include* (`<jsp:include page="url"/>`). Debido a que el botón salir se incluye en varias páginas de aplicación su fragmento de código se ha extraído a un fichero *JSP* independiente, evitando así la redundancia de código.

NOMBRE	FUNCIONALIDAD
<i>boton_salir.jsp</i>	Muestra el botón de salir de la aplicación.

Tabla 3.11.- Fichero JSP del directorio *varios*

El directorio **WEB-INF** contiene el fichero *web.xml*. Este fichero configura la aplicación. Es donde se han declarado los servlets. Además del fichero *web.xml* tiene 2 directorios. El directorio *classes* y el directorio *lib*.

- El directorio **classes** es en el que están los ficheros compilados de las clases (incluye los *servlets*) utilizadas por la aplicación web.

NOMBRE	FUNCIONALIDAD
<i>servlet_nombresjugadores.class</i>	Obtiene la relación de los nombres de los jugadores registrados en el juego y los ordena alfabéticamente.
<i>servlet_redirigiropcion.class</i>	Redirige al <i>servlet</i> correspondiente en función de la acción que haya seleccionado el usuario en <i>seleccionajugador.jsp</i> (eliminar o consultar resultados)
<i>servlet_resultadosjugador.class</i>	Extrae de la base de datos las puntuaciones del jugador seleccionado.
<i>servlet_eliminarjugador.class</i>	Elimina de la base de datos el registro correspondiente al jugador seleccionado y vuelve a obtener la relación actualizada de los jugadores registrados en el juego.
<i>servlet_finjuego.class</i>	Invalida la sesión del usuario que sale de la aplicación.

Tabla 3.12.- Ficheros *servlet* del directorio *classes*

- Directorio **lib** contiene la biblioteca (comprimida con *jar*) *mysql-connector-java-5.1.12-bin* que utiliza la aplicación.

4. EVALUACIONES

Como se ha comentado en la sección 2.4., durante el proceso de diseño de la interfaz se contó con la ayuda de profesionales docentes. Estos profesionales fueron profesores participantes en el programa PROMENTOR, quienes trabajan diariamente con personas con discapacidades. Gracias a su ayuda, el diseño de la interfaz de este juego fue evolucionando desde los prototipos iniciales hasta la versión final.

PROMENTOR es un programa específico donde usuarios con discapacidades cognitivas aprenden a llevar a cabo tareas relacionadas con puestos de trabajo reales a través de la que ellos deben finalmente obtener una inserción laboral. Algunas de estas tareas son clasificación de documentos o envío y recepción de emails. Además, en este programa aprenden conceptos relacionados con actividades diarias y realizan actividades de *role-playing* con el objetivo de mejorar su autonomía y su habilidad para tomar decisiones. PROMENTOR surgió de un acuerdo entre la Universidad Autónoma de Madrid y la fundación PRODIS, y su objetivo final es capacitar a personas con discapacidades cognitivas para una inserción laboral, para lo cual es indispensable que estas personas consigan manejarse con independencia en la red de transporte público de la Comunidad de Madrid logrando ir de su casa a su lugar de trabajo.

Una vez que el juego se terminó de implementar, se realizó una segunda evaluación por cuatro profesionales del proyecto BIT de la Fundación Síndrome de Down de Madrid [31]. El proyecto BIT es un proyecto de I+D+i de carácter educativo y tecnológico, cuyo principal objetivo es acercar el uso de las Tecnologías de la Información y la Comunicación (TIC) a las personas con síndrome de Down y/o discapacidad intelectual con el fin de abrirles nuevas vías para su integración social, educativa y laboral.

La evaluación realizada consistió en una breve exposición de los objetivos del juego y en mostrar el funcionamiento del mismo. La dinámica del juego que se expuso es similar al manual breve para docentes que aparece en el apéndice 7.2. A continuación de la explicación, se realizó una breve encuesta cuyas preguntas se pueden consultar en el apéndice 7.3. Las principales conclusiones extraídas de la encuesta realizada se exponen a continuación:

- La interfaz es simple e intuitiva para las personas con discapacidad cognitiva. Los elementos mostrados en la misma tienen un tamaño adecuado.
- La estructuración del juego (pantallas mostradas (presentación de la ronda, presentación de cada pregunta junto con su retroalimentación a continuación, resultados obtenidos en la ronda, visión general en los distintos niveles del juego) es apropiada.
- Se ha cuidado una retroalimentación correcta y los mensajes motivadores que ayuden a los usuarios a continuar su entrenamiento.
- La aplicación desarrollada servirá a los profesionales para poder entrenar a las personas con discapacidad cognitiva en el uso de la red de metro de Madrid antes de realizar prácticas in-situ. Este hecho podrá favorecer la adquisición de aprendizajes y que el tiempo empleado en la fase de entrenamiento en el mundo real se reduzca.

Aparte de estas conclusiones, los profesionales de la Fundación Síndrome de Down de Madrid sugirieron una serie de mejoras sobre el juego con el objetivo de facilitar una mayor comprensión a las preguntas del juego y una forma de trabajar más flexible. Estas mejoras se presentan a continuación:

- Inclusión de audio junto con el texto del enunciado en las preguntas formuladas por el juego. Aunque las preguntas están adaptadas para que tengan una lectura fácil, algunos usuarios tienen muchas dificultades en la comprensión del texto. El hecho de proveer la información por dos canales facilita la comprensión del texto leído.
- Inclusión de más niveles en el juego. Nos indicaron que en el caso concreto de las preguntas que realiza el juego donde los usuarios tienen que escribir la respuesta, un nivel superior sería que se siguiera ayudando a la hora de reconocer el nombre correcto de la estación (p.e. “¿Quisiste decir Príncipe de Vergara?” en la figura 2.20) aunque obligando al usuario a que escribiera el nombre de la estación correctamente en la correspondiente caja de texto. Además, los botones de sí/no desaparecerían de la parte de la interfaz gráfica del juego en este tipo de preguntas.
- Inclusión de una pantalla intermedia en el caso de que el usuario falle una pregunta. En este caso, habría que pedir al jugador que vuelva a intentar de nuevo otra vez la pregunta. Esta pantalla intermedia se mostraría al usuario para darle una segunda oportunidad para acertar esta pregunta.
- Aunque en general los textos son de lectura fácil, es necesario revisar aquellos que corresponden a preguntas de nivel superiores ya que alguno tiene demasiada longitud y es necesario acortarlo.
- Realización de una herramienta de autor que permitiera a los profesionales la creación e inclusión de preguntas relacionadas con una determinada zona del plano del metro de Madrid o con una o varias líneas. El objetivo de esta herramienta sería que en los primeros niveles del juego se trabajase con una zona más concreta centrándose en los trayectos para ir de su hogar hasta el lugar de trabajo.

5. CONCLUSIONES Y POSIBLES TRABAJOS FUTUROS

5.1. LOGROS ALCANZADOS

Este proyecto fin de carrera ha conseguido realizar una aplicación adaptada al nivel de conocimiento de los usuarios cuyo objetivo es que usuarios con discapacidades intelectuales puedan utilizar de forma independiente el transporte público. El juego permite que exista un entrenamiento previo de las personas con discapacidad antes de realizar pruebas reales con técnicos de apoyo en el metro de Madrid. Con esta aplicación los profesionales pueden ahorrar tiempo empleado en el entrenamiento de desplazamientos.

La aplicación ha sido validada tanto en la fase de diseño como en la fase de implementación por profesionales obteniendo muy buenos resultados gracias a su asesoramiento. El diseño de la aplicación es de fácil uso, motivador, atractivo y de carácter lúdico para lograr, captar la atención e interés de los usuarios hacia el juego desde un primer momento. La interacción con actividades lúdicas motiva a los usuarios a llevar a cabo las actividades. Se ha adaptado la interacción del jugador con la aplicación para mitigar los problemas que presentan de motricidad fina. Con el objetivo de facilitar la lectura y comprensión se ha utilizado un tipo y tamaño de letra validado por profesionales. Los colores de las letras resaltan del fondo sobre el que se encuentran y se prima el uso del canal visual. En todo momento, se proporciona la máxima información de lo que se debe hacer mediante instrucciones claras y fáciles de leer. Se ha aprovechado la secuenciación que permite el ordenador para lograr la focalización de la atención en los elementos relevantes en cada momento.

El juego tiene una amplia variedad de preguntas para evitar que surja en los jugadores sentimiento de cansancio y desmotivación que llevan irremediablemente a mostrar una falta de interés por la actividad y su posterior abandono. Además, la disposición de las respuestas posibles que se dan al usuario tiene una ordenación aleatoria, para evitar que el jugador esté predispuesto a seleccionar siempre la respuesta localizada en la misma área. Se ofrecen ayudas en las preguntas en las que el usuario puede cometer errores de escritura debido a sus dificultades en el proceso de lecto-escritura. De esta manera se evita que los usuarios no se sientan frustrados. Se utilizan reforzadores que son siempre positivos para lograr motivarles a continuar jugando y no crear sentimiento de desánimo.

Las preguntas se han planteado (cuando ha sido posible) dentro de distintos y múltiples contextos de la vida cotidiana para lograr una posterior generalización y transferencia de los aprendizajes (tanto de contenidos como de habilidades) a dichos contextos de la vida cotidiana. Para ayudar en la consolidación de aprendizajes y mejorar los resultados de memoria se ha adaptado el juego para que las cuestiones se presenten de forma reiterada. Con éstas y otras adaptaciones se pretende ofrecer al usuario representaciones lo más reales posibles que ayuden a su posterior representación mental. Todo ello encaminado a diseñar condiciones y situaciones de aprendizaje orientadas hacia su aplicación en la vida cotidiana.

Por último, es importante hacer notar que el trabajo presentado en este Proyecto Fin de Carrera ha sido publicado en categoría de póster (ver Figura 5.3) dentro del congreso internacional “*User Modeling, Adaptation and Personalization*” (UMAP 2010) incluido en el CORE. UMAP es la principal conferencia internacional para investigadores y profesionales que trabajan en sistemas que se adaptan a usuarios individuales o a grupos de usuarios. El trabajo titulado “*An Adaptive Game to Train Users with Special Needs to Make Decisions: Using Public*

"Transportation" se puede encontrar en los *Adjunct Proceedings of UMAP 2010*. Puede descargarse de la siguiente dirección: http://web41.its.hawaii.edu/www.hawaii.edu/UMAP2010/downloads/UMAP2010_AdjunctProceedings.pdf.

An Adaptive Game to Train Users with Special Needs to Make Decisions: Using Public Transportation

Estefanía Martín, Silvia Calvo

Escuela Técnica Superior de Informática, Universidad Rey Juan Carlos, Spain
estefania.martin@urjc.es, s.calvog@alumnos.urjc.es

Rosa M. Carro

Escuela Politécnica Superior, Universidad Autónoma de Madrid, Spain
rosa.carro@uam.es



Motivation

During our life, we are involved in a huge variety of activities and have to make many decisions. However, not all the users have the same capabilities. Decision-making, as well as the tasks to be performed daily, can be trivial for some users and quite hard for others.

The adaptive game presented should contribute to train users with cognitive limitations to use public means of transport. This game attempts to increase user personal autonomy and to improve abilities such as user self-control and decision-making.

User Model

- Individual cognitive functioning
- Verbal Comprehension Index
- Perceptual Reasoning Index
- Working Memory Index
- Processing Speed Index
- Level of knowledge

The screenshot displays the game's user interface with several levels and questions. The interface includes a character icon, question text, and interactive elements like buttons and progress indicators.

LEVEL 4 - QUESTION 6: Which underground stations are missing in the picture? (Estrella, Station 1, O'Donnell, Manuel Becerra, Station 2, El Carmen)

LEVEL 4 - QUESTION 4: I am travelling from La Granja station (line 10) to Plaza de Castilla, where I am meeting a friend. Suddenly, the train breaks at Chamartín station. Which is the most suitable line to get my destination?

LEVEL 1 - QUESTION 1: Which is the color of line 4 in Madrid subway?

LEVEL 2 - QUESTION 7: Which is the picture with one station missing? (Hortaleza, San Lorenzo, Mar de Cristal, Pinar del Rey, Palos de la Frontera, Delicias, Legazpi, Usera)

LEVEL 2: SILVIA, your score is 87 points at level 2. Points: 87

LEVEL 1: Points: 91

LEVEL 2: Points: 87

LEVEL 3: Points: 83

LEVEL 4: Points: 50

Figura 5.3.- Póster publicado en el congreso internacional "User Modeling, Adaptation and Personalization" (UMAP 2010).

5.2. POSIBLES TRABAJOS FUTUROS

En el futuro próximo, se va a evaluar el juego con usuarios con discapacidades cognitivas tanto del programa PROMENTOR como de la Fundación Síndrome de Down de Madrid. Esperamos obtener buenos resultados, ya que expertos tanto del programa PROMENTOR como de la Fundación Síndrome de Down de Madrid han colaborado con nosotros en el análisis de requerimientos y en el diseño del juego, y lo han validado una vez que se ha desarrollado.

Como ya se ha comentado en el apartado anterior, el juego está adaptado a personas con discapacidades cognitivas. Sin embargo, podrían realizarse las mejoras propuestas por el equipo de profesionales de la Fundación Síndrome de Down de Madrid. Aparte de estas mejoras, se propone tres mejoras adicionales que consisten en:

- Complementar la aplicación para el seguimiento de los resultados obtenidos por los usuarios incluyendo en la interfaz gráficos que representen la evolución del usuario en el tiempo y que ayuden al profesional a la hora de detectar los aspectos que se necesitan trabajar.
- Modificación del concepto de camino mínimo, incluyendo los tiempos entre estaciones. Esto implicaría disponer de los tiempos que hay entre todos los pares de estaciones, así como los tiempos de transbordo entre líneas, aunque estos dos aspectos podrían variar dependiendo de la hora del día.
- Inclusión tanto la estimación global de las funciones cognitivas individuales como las cuatro puntuaciones de los índices que representan los principales componentes de inteligencia como rasgos de adaptación del juego. Esta información será extraída a partir de los datos obtenidos con el cuestionario WAIS III [32]. De esta manera el juego no tendría solamente en cuenta el nivel de conocimiento del usuario sino que podría utilizar estos nuevos indicadores para realizar adaptaciones y adecuar el funcionamiento del mismo en base a las características personales de cada usuario.

6. BIBLIOGRAFÍA

1. Vera L., Llorens C., Martínez B., Sebastián B., Casas X., Herrera G.. "Herramienta multimedia para la preparación en empleo con apoyo en personas con Síndrome de Down".
2. Carmien S., Dawe M., Fischer G., Gorman A., Kintsch A., Sullivan J. F.: Sociotechnical environments supporting people with cognitive disabilities using public transportation. ACM Transactions on Computer Human Interaction, vol. 12(2), pp. 233-262. ACM. New York, USA (2005)
3. Pérez Sánchez L. F., Beltrán Llera J. y Sánchez Pozuelo E. (Universidad Complutense de Madrid). "Un programa de entrenamiento para la mejora de los déficits de memoria en personas con síndrome de Down". Psicotherma 2006. Vol. 18, nº 3, pp. 531-536. <http://www.psicotherma.com>
4. Henao Álvarez, O., Ramírez Salazar D. A. y Giraldo López L. E. "Una propuesta didáctica apoyada en tecnología multimedial para el desarrollo de habilidades comunicativas en niños con síndrome de Down". Universidad de Antioquía, Facultad de Educación.
5. González C., Sigut J., Sanabria H., Guerra D., Noda M., Bruno A., Hernández B., Hernández A., Moreno L.. "Diseño e implementación de interfaces accesibles para acercar las matemáticas a niños con síndrome de Down." Current Developments in Technology-Assisted Education : 1090-1095, 2006.
6. Ortega, J.M, Gómez-Ariza, C.J. "El multimedia en el aprendizaje de las matemáticas en niños con Síndrome de Down". EDUTEC 2004. Barcelona.
7. Gómez-Ariza, C.J. "Nuevas tecnologías y aprendizaje matemático en niños con Síndrome de Down: Generalización para la autonomía". Pixel-Bit. Revista de Medios y Educación, enero 2007, número 029. Universidad de Sevilla. Sevilla, España, pp 59-72.
8. <http://amilab.ii.uam.es/doku.php?id=fling>
9. <http://www.ceiec.es/site/index.php?id=58>
10. Investigador Principal: Felipe Ibarrarán Cobo. "Desarrollo e integración de una herramienta de generalización y gestión de tareas para optimizar un sistema automatizado de ayuda dirigido al colectivo laboral de personas con Síndrome de Down". TASKTOOL. Proyecto 156/05. Gureak Araba S.L. (2005-2006).
11. González J. L., Gutiérrez F. L., Cabrera M. J.. "Diseño de videojuegos colaborativos adaptados a la Educación Especial". Dpto. Lenguajes y sistemas Informáticos. ETS Ingenierías Informática y de Telecomunicación. Universidad de Granada.
12. "Los jóvenes con discapacidad en España. Informe de situación 2010". Colección Barclays Igualdad y Diversidad, número 4. Grupo Editorial Cinca. Abril 2010.
13. Pascual Martínez Gómez A. "Las personas con Síndrome de Down en España. Perfil estadístico". Federación Española del Síndrome de Down (FEISD). Documentación Social 120: pp 207-227. 2003.

14. Ortega, J.M. "Bondades y limitaciones del Material Multimedia para personas con síndrome de Down". Revista Síndrome de Down 22: 84-92, 2005.
15. Arregi Martínez A. "Síndrome de Down: necesidades educativas y desarrollo del lenguaje". Gobierno Vasco. Departamento de educación, universidades e investigación. Instituto para el Desarrollo Curricular y la formación del profesorado. Área de Necesidades Educativas Especiales. 1997.
16. Pavón Mestras J. Estructura de las Aplicaciones Orientadas a Objetos. El patrón Modelo-Vista-Controlador (MVC). Facultad de Informática. Dep. Ingeniería del Software e Inteligencia Artificial. Universidad Complutense Madrid
17. Bascón Pantoja E. El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing. Acta Nova; Vol. 2, Nº 4, diciembre 2004.
18. <http://es.wikipedia.org/wiki/html>
19. Gómez Basanta J. J. "Introducción a HTML y JavaScript". 1997.
20. Eguíluz Pérez J. "Introducción a JavaScript". <http://www.librosweb.es>. 2009.
21. <http://www.w3c.es/divulgacion/guiasbreves/hojasestilo>
22. García de Jalón J. y otros. "Aprenda Java, como si estuviera en primero", Campus tecnológico de la Universidad de Navarra. Escuela Superior de Ingenieros. San Sebastián, Enero 2000.
23. García de Jalón J. y otros. "Aprenda Servlets de Java como si estuviera en segundo", Universidad de Navarra. Escuela Superior de Ingenieros Industriales. San Sebastián, Abril 1999.
24. Polo M. y Villafranca D. "Introducción a las aplicaciones Web con Java". Universidad de Castilla-La Mancha. Escuela Superior de Informática. 2007.
25. Sánchez J. "Java2", 2004.
26. Manual de referencia de MySQL: <http://dev.mysql.com/doc/refman/5.0/es/index.html>
27. GIMP: <http://www.gimp.org.es>
28. PowerPoint: <http://office.microsoft.com/es-es/powerpoint/HA101656373082.aspx>
29. Página oficial del entorno de desarrollo NetBeans: <http://www.netbeans.org/>
30. Página oficial del servidor de aplicaciones de Sun GlassFish: <http://glassfish.org/>
31. Fundación Síndrome de Down de Madrid: <http://www.downmadrid.org/>
32. Wechsler, D.: The Wechsler Adult Intelligence Scale. San Antonio, TX: Psychological Corporation (1997).

7. APÉNDICES

7.1 MANUAL DE INSTALACIÓN

Para poner en funcionamiento las aplicaciones Web desarrolladas antes es necesario tener instaladas una serie de aplicaciones.

En primer lugar instalar *Java Development Kit 6 (JDK6)*, se encuentra disponible para su descarga en el enlace <http://www.oracle.com/technetwork/java/javase/downloads/jdk6-jsp-136632.html>.

A continuación instalar *Glassfish v.3*. Se puede encontrar para su descarga gratuita en el enlace <http://glassfish.java.net/es/downloads/3.0.1-final.html>, es el contenedor de *servlets* donde será desplegada la aplicación Web.

Por último proceder a instalar *MySQL 5.5* (disponible en <http://www.mysql.com/downloads/mysql/>) que almacenará la base de datos necesaria para el correcto funcionamiento de la aplicación. El mismo programa de instalación se encarga de guiar al usuario durante el proceso. Una vez instalado es necesario crear la base de datos “*jugadores*” e importar la base de datos a partir del archivo “*jugadores.sql*” que se facilita junto con la documentación del proyecto para la creación de la tabla y de la base de datos.

En este punto, abrir los ficheros “*configuración acceso bd.txt*” que se encuentran en “*MetrodeMadrid/web/vari*os” y en “*SeguimientoResultados/web/vari*os” y sustituir las 3 líneas que aparecen por el usuario y contraseña de acceso al servidor de *MySQL*, así como el servidor de la aplicación respectivamente. Ahora ya se pueden abrir los proyectos con *NetBeans IDE* para generar los ficheros *.war* de ambas aplicaciones desarrolladas.

Una vez instaladas correctamente las aplicaciones, importada la base de datos, configurado los ficheros “*configuración acceso bd.txt*” y haber generado los ficheros *.war*, tan sólo debemos arrancar el Servidor de Aplicaciones (*Start Application Server*) de *GlassFish*, acceder a la consola de administración (*Administration Console*) de *GlassFish* y en la sección de *Aplicaciones* pulsar sobre el botón *Implementar* para buscar el archivo *MetrodeMadrid.war*, en el cuadro de texto *Bibliotecas* escribir */lib/mysql-connector-java-5.1.12-bin* y pulsar sobre el botón *Aceptar*. Volver a pulsar el botón *Implementar* y realizar el mismo procedimiento con el archivo *SeguimientoResultados.war*.

Ahora sólo se tiene que pulsar sobre la opción *Iniciar* correspondiente a la aplicación que deseemos ejecutar.

7.2. MANUAL BREVE PARA DOCENTES

⇒ Estructura del juego

El juego está estructurado en un sistema de 4 niveles de creciente dificultad. El juego consiste en rondas, de 10 preguntas cada una, asociadas a cada nivel, que tienen que ser contestadas por los jugadores. El juego permite al usuario jugar tantas rondas como desee de los niveles a los que tiene acceso en cada momento.

Al inicio del juego los jugadores sólo tienen acceso al nivel 1, para poder tener acceso al nivel inmediatamente superior (el nivel 2 en este caso) deben lograr un porcentaje mínimo de respuestas correctas del 80% de las preguntas planteadas en el nivel superior al que tienen acceso (el nivel 1 si es el comienzo del juego) y así sucesivamente hasta que logren desbloquear el acceso a los 4 niveles del juego. Si al jugar una ronda de un nivel ya superado obtiene una puntuación para ese nivel menor al 80% se le desactiva el acceso a los niveles superiores.

⇒ Cómo jugar

Para comenzar el juego pulsamos sobre el botón “Empezar”. Si es la primera vez que jugamos pulsamos el botón “Nuevo Jugador” y escribimos el nombre con el que queremos registrarnos en el juego, si ya hemos jugado en alguna ocasión al juego pulsamos sobre el botón “Continuar Partida”, y en la pantalla que se muestra escribimos el nombre con el que estamos registrados para acceder a nuestro perfil.

Una vez que hemos accedido al juego, si es la primera vez que jugamos comienza directamente con una ronda de 10 preguntas del nivel 1. Si hemos jugado en otras ocasiones se muestra una pantalla con la puntuación que tenemos en cada nivel (ver Figura 7.1) de los que tenemos acceso. Para jugar una ronda pulsamos sobre el botón “Empezar” del nivel al que deseamos jugar, así aparece el enunciado de la primera pregunta de la ronda.



Figura 7.1. Pantalla de inicio de ronda

Con cada pregunta se muestran dos o más posibles respuestas y tenemos que seleccionar la correcta. Además en el nivel de mayor dificultad (*nivel 4*), hay preguntas en las que el jugador tiene que escribir mediante el teclado su/s respuesta/s a la pregunta (ver Figura 7.2). Cada pregunta tiene una única respuesta correcta asociada, si bien hay en algunos tipos de pregunta que la respuesta está compuesta de 2 o más respuestas (ver Figura 7.3).

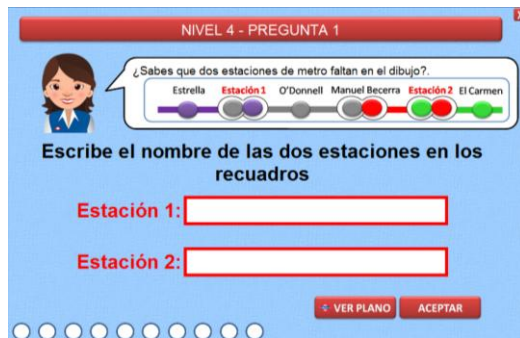


Figura 7.2. Tipo de pregunta de texto libre



Figura 7.3. Tipo de pregunta de selección

Cuando el jugador quiere seleccionar una respuesta tiene que pulsar sobre ella, e inmediatamente ésta aparece en el área de respuesta. Si queremos cambiar la respuesta dada pulsamos sobre la nueva respuesta que queremos seleccionar y ésta aparecerá en el área de respuesta sustituyendo a la respuesta anterior. Si lo deseamos podemos consultar el plano del Metro de Madrid pulsando sobre el botón “Ver Plano”.

Para conocer si nuestra respuesta es correcta pulsamos sobre el botón “Aceptar” y se nos muestra una pantalla de retroalimentación (ver Figura 7.4).

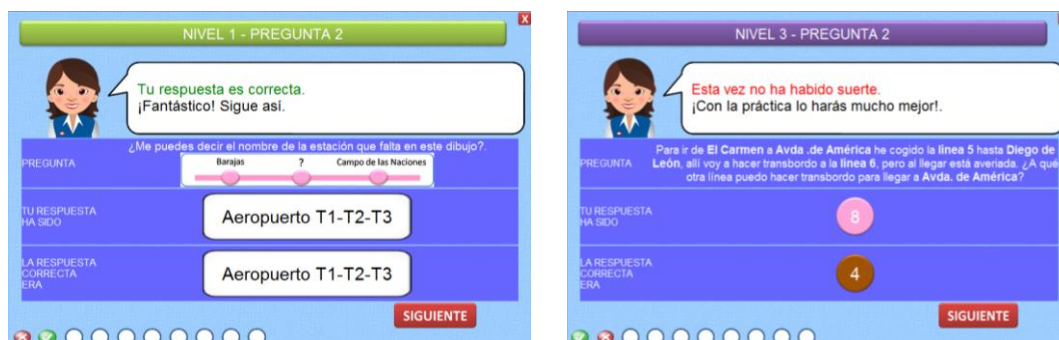


Figura 7.4. Ejemplos de pantallas de retroalimentación a diferentes tipos de preguntas

Una vez comprobada la respuesta pulsamos sobre el botón “Siguiente” para contestar a la siguiente pregunta de la ronda. Si ya hemos contestado a las 10 preguntas de la ronda se muestra una pantalla con la puntuación obtenida en el nivel una vez contestadas las preguntas de la ronda actual (ver Figura 7.5).



Figura 7.5. Pantalla con el resumen de puntos obtenidos después de jugar una ronda

Al pulsar sobre el botón “Siguiente” se vuelve a la pantalla con el resumen de puntuaciones actualizado después de jugar la última ronda. Si obtenemos más de 80 puntos en

el nivel superior al que teníamos acceso se habilita el acceso al siguiente nivel (ver Figura 7.6). Para comenzar una nueva ronda de preguntas tan sólo tenemos que volver a pulsar sobre el botón “Empezar” del nivel al que deseemos jugar.



Figura 7.6 Pantalla con el resumen de las puntuaciones obtenidas en todos los niveles

Para salir del juego en cualquier momento, podemos pulsar sobre el botón con un dibujo de “x” en su interior situado en la parte superior derecha de todas las pantallas.

7.3. CUESTIONARIO DE EVALUACIÓN DEL JUEGO “METRO DE MADRID”

Instrucciones: A continuación, se formulan una serie de preguntas. Las preguntas pueden calificarse de 1 a 5 puntos siendo 1 la calificación más baja (en absoluto de acuerdo) y 5 la calificación más alta (totalmente de acuerdo). Agradecemos de antemano la sinceridad en sus repuestas y sus comentarios o sugerencias sobre este juego.

	1	2	3	4	5
1. La interfaz es simple y apropiada	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2. La interfaz es intuitiva	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. Los elementos de la pantalla tienen un tamaño adecuado	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4. Los elementos de la pantalla tienen una disposición apropiada	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5. La secuencia de las pantallas mostradas en las rondas (presentación general, presentación de cada pregunta junto con su retroalimentación, resultados obtenidos en la ronda, visión general en los distintos niveles del juego) es apropiada	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6. La estructuración en los 4 niveles que tiene el juego me ha parecido apropiada	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7. Las preguntas que se realizan en cada uno de los niveles son apropiadas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8. El nivel de dificultad se incrementa adecuadamente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9. ¿Crees que la aplicación desarrollada ayudará al conocimiento de la red de Metro de Madrid?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10. ¿Crees que la aplicación desarrollada ayudará al usuario en la toma de decisiones en situaciones habituales ?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11. ¿Crees que la aplicación desarrollada ayudará al usuario en la toma de decisiones en situaciones inesperadas?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12. Los refuerzos incluidos en el juego son adecuados para motivar al jugador	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13. El lenguaje utilizado en el juego es apropiado para usuarios con síndrome de Down	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14. La forma en que el sistema le permite realizar las tareas solicitadas al usuario es apropiada	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15. La navegación a través del sistema resulta sencilla	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16. En general la apariencia del juego es apropiada	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	COMENTARIOS
17. ¿Qué aspectos del usuario crees que se pueden tener en cuenta a la hora de adaptar el funcionamiento del juego aparte del nivel de conocimiento del usuario en cada uno de los tipos de pregunta?	
18. ¿Qué aspectos del usuario crees que se pueden tener en cuenta a la hora de adaptar la interfaz del juego (p.e. eliminación de elementos que aparecen en las pantallas, modificación de los mismos, añadir más elementos, etc.)?	
19. Destaca los aspectos que te han parecido más negativos de la aplicación	
20. Destaca los aspectos que te han parecido más positivos de la aplicación	
21. Comentarios y sugerencias sobre la aplicación	