



Universidad Complutense de Madrid

Facultad de Informática

Proyecto de Sistemas Informáticos

CS – Training

Autores

Alejandro Palacios Provencio

Roberto Tornero Santamarina

Profesores Directores

Pablo Moreno Ger

Baltasar Fernández Manjón

Año académico 2008 / 2009

1.	Tags	5
2.	Resumen	5
	Castellano	5
	English	5
3.	Introducción.....	6
	Motivación.....	6
	Nintendo DS y un LMS - Integración y Comunicación.....	7
	Objeto de Estudio	7
	Papel de los juegos en la educación.....	8
	Inspiración en Brain Training.....	9
	e-Learning y LMS.....	12
	Learning Management Systems	13
	Desarrollo de este documento	15
4.	Modelo General.....	16
	Ejercicios Interactivos o MiniJuegos.....	16
	Modos de uso.....	17
	Práctica Libre o "Entrenamiento" (no supervisado).....	17
	Test Evaluados o "Evaluación" (supervisado).....	18
	Participación del profesor	18
	Puntuación evolutiva.....	19
	Posibles campos de aplicación	19
5.	CS – Training.....	21
	Descripción General.....	21
	Ámbito Nintendo DS	22
	Ámbito LMS – Moodle	23
	Ejercicios Interactivos (E.I.).....	23
	Elementos de interacción	24
	Elementos de control.....	24
	Elementos de contenido.....	25
	Ejercicio Interactivo 1 - Puzzle de Programación.....	30
	Ejercicio Interactivo 2 - Recorrido de árboles	33
	Ejercicio Interactivo 3 - Visibilidad de variables	35
	Ejercicio Interactivo 4 – Árboles Mini-Max.....	37
	Ejercicio Interactivo 5 – Programación ROM / PAL.....	40

	Ejercicio Interactivo 6 – Piezas lógicas	42
	Ejercicio Interactivo 7 – Recorrido de Circuitos Sencillos	44
	Ejercicio Interactivo 8 – Mapas de Karnaugh	46
	Ejercicio Interactivo 9 – Recorrido de intensidades	49
	Ejercicio Interactivo 10 – Herencia y Polimorfismo	51
6.	Arquitectura General	54
	Visión de Alto Nivel (Aplicación)	55
	Diagrama de clases (UML).....	56
	Cargador.....	56
	Ejercicio Interactivo.....	57
	Evaluador.....	58
	Esquemas Concretos	58
	Aplicación de Nintendo DS.....	59
	Comunicación e Integración con un LMS.....	59
7.	Proceso de desarrollo	62
	Análisis inicial.....	62
	Estudio del dominio.....	62
	Análisis de riesgos.....	62
	Organización del modelo de proceso.....	63
	Métricas del proyecto.....	64
	Desarrollo en iteraciones.....	64
	Primera Iteración	65
	Segunda Iteración	67
	Tercera Iteración.....	71
8.	Conclusiones y Trabajo Futuro.....	73
	Herramienta de Autoría.....	73
	Creación de una Nueva Instancia con la Herramienta de Autoría	74
	Ejercicios Interactivos en Moodle	75
9.	Referencias	77
10.	APENDICES	78
	APENDICE A	78
	Librerías PALib.....	78
	Textos.....	78
	Uso del Pad.....	79
	Uso del Stylus.....	80
	Uso del teclado.....	80

Mostrar y desplazar imágenes	81
Rotación y zoom de imágenes	82
Transparencias y Profundidad en las pantallas	82
Funciones matemáticas importantes para juegos	83
Programación en 3D	83
Comunicación	84
La Integración con Moodle	85
Configuración de Moodle	85
Comunicación con Moodle.....	87
Cómo se hace en la Nintendo DS.....	88
Otras Librerías	89
APENDICE B.....	90
Manual del Profesor.....	90
Cadena de configuración de examen.....	90
Lógica de Evaluación.....	91

1. Tags

CS Training, NDS, Moodle, LMS, e-learning, Brain Training, educativo, integración

2. Resumen

Castellano

Partiendo de los conocimientos adquiridos en la carrera de Ingeniería Informática y de una idea inspirada en el juego "Brain Training", nace CS-Training, una novedosa aplicación educativa que permite a los estudiantes, por primera vez, reforzar los conocimientos de las materias que estudian en la palma de su mano, ¡y en horario de clase! Añadiendo una integración completa de la aplicación en los entornos e-learning habituales, con el soporte de los Sistemas de Gestión del Conocimiento, e involucrando con una larga lista de funciones a los instructores, CS-Training se alza como el primer paso en el camino hacia la inserción completa de los videojuegos educativos en la enseñanza universitaria actual.

English

Based on the acquired knowledge in Computer Engineering and an idea inspired by the videogame "Brain Training", CS-Training is born as an innovative application that allows students to enhance, for the first time, their knowledge of the subjects they study in the palm of their hand, and while they're in class time! Adding a complete integration of the application in the usual e-learning environments, supported by the Learning Management Systems, and involving a long list of functions to the instructors, CS-Training stands as the first real step into the insertion of educational video games at the present university learning system.

3. Introducción

Ante el desarrollo de un nuevo proyecto de Sistemas Informáticos, una idea rondaba nuestras cabezas: de qué manera podíamos contribuir en la mejora de los procesos de enseñanza y aprendizaje con ayuda de los videojuegos educativos y las Tecnologías de la Información y Comunicación. En este ámbito ya se contaba con un gran conjunto de infraestructuras y aplicaciones, por lo que una propuesta poco innovadora no tendría la repercusión que se pretendía. Frente a tal reto surgió CS-Training, una nueva forma de entender la integración de las plataformas portátiles de videojuegos en entornos de enseñanza electrónica.

CS-Training permite entrenar un amplio abanico de ejercicios prácticos de las asignaturas que se imparten en la Ingeniería Informática desde la videoconsola Nintendo DS, y cuenta con el soporte completo de un Sistema de Gestión del Aprendizaje, creando un binomio que establece una nueva modalidad de enseñanza interactiva en las aulas, siguiendo los estándares e-learning más comunes. Una idea fresca e innovadora, que tiene su mayor atractivo en la posibilidad de adaptarse fácilmente a los modelos de enseñanza actuales, proponiendo un modelo mixto en el que profesores y alumnos jugarán un papel determinante.

A lo largo de este documento conoceremos los entresijos de la realización de CS-Training, comenzando en primera instancia por aquello que nos llevó a realizar un proyecto de estas características.

Motivación

Los videojuegos son una forma de ocio electrónico que ha alcanzado un éxito notable en estas últimas décadas. Concretamente, los videojuegos educativos obtienen unas cifras de ventas muy elevadas gracias a la sencillez de su propuesta y lo atractivo de su desarrollo, abarcando un espectro de públicos muy amplio. Es difícil encontrar actualmente adolescentes que no conozcan el medio, o incluso que no posean una videoconsola, lo que permite una toma de contacto casi directa entre los juegos de corte educativo y estos adolescentes. Razones para su éxito pueden ser desde las destrezas que se adquieren, hasta los conocimientos que se pueden obtener en muchas de las áreas de aplicación.

Dentro de las distintas posibilidades de aprender mediante una forma de ocio, ya sea mediante Internet o un ordenador personal, es remarcable el aumento de uso que han adquirido los dispositivos portátiles. Videoconsolas, reproductores de música con diversas opciones, ordenadores portátiles o incluso, teléfonos móviles, gozan de una cantidad considerable de aplicaciones enfocadas al aprendizaje electrónico, algunas de un éxito inconcebible hasta hace unos años. Entre estos dispositivos, sobresale especialmente la plataforma Nintendo DS, con el fenómeno del videojuego "Brain Training: Entrena tu Cerebro" a la cabeza. Mientras que en este

videojuego se ejercitan capacidades de cálculo rápido, memoria, vista, nuestra mayor motivación consistía en ir un paso más allá: adaptar los conocimientos de la enseñanza actual impartida en facultades, a una aplicación e-learning sobre la base de Brain Training, usando como plataforma el portátil de Nintendo. Pero de habernos estancado en esta idea, estaríamos adaptando el formato de un videojuego a otro videojuego, que aún siendo de corte educativo, necesitaba de alguna característica más relevante si se quería causar un gran impacto.

Nintendo DS y un LMS - Integración y Comunicación

La posibilidad de dar un soporte educativo completo a la aplicación integrando la funcionalidad de un Sistema de Gestión del Aprendizaje abre nuevos horizontes en el terreno de los videojuegos educativos portátiles. Se trataba de crear una arquitectura novedosa para conseguir una integración total, distinguiendo diversas vías de comunicación y ampliando de manera más que significativa la experiencia e-learning del conjunto. No existen precedentes en este ámbito en los que poder inspirarse, puesto que en su totalidad, los videojuegos educativos con base en Nintendo DS se limitan a la plataforma como tal, sin involucrar otros medios de aprendizaje. Estos límites los hacen ineficaces para tratarlos como modelo mixto en las aulas.

Sin embargo, CS-Training se sustenta sobre los muchos servicios y recursos que brindan los LMS. Entre estos servicios o posibilidades que ofrecen los LMS encontramos características fundamentales como la administración y gestión de las materias de los cursos, la comunicación en casi todo momento entre profesores y alumnos, rankings globales en los que medir las capacidades de cada alumno y reforzar la vertiente competitiva de la aplicación, etc.

Es un campo aún por explotar debidamente, pero se pretende dar un primer gran paso en lo que, sin lugar a dudas, podría revolucionar la manera de aprender de los adolescentes en las aulas.

Objeto de Estudio

Nos hemos decantado por el campo de aprendizaje de la Informática. Las razones de la elección de este campo de conocimiento son debidas, principalmente, a que contamos con los recursos y las fuentes necesarias para llevar a cabo todo tipo de ejercicios y exámenes relacionados con la materia.

Dentro del campo de la Informática se quiere profundizar en todos los aspectos posibles, esto es, en los ámbitos software y hardware. En el primero, se busca un recorrido desde la programación más básica, hasta las estructuras de datos, tipos de algoritmos y su funcionamiento y aspectos de ingeniería del software. En el segundo, la implementación de distintos tipos de sistemas, construcciones de funciones de conmutación y teorías acerca de circuitos.

Pero el hecho de trasladar los ejercicios prácticos que se imparten en las aulas no nos garantiza que produzcan la motivación necesaria sobre los alumnos. Se buscaba la manera de convertir estos ejercicios en un reto, que fomente la sana competencia que hay presente en todo videojuego. Combatimos la posible pereza del estudio de las asignaturas con el añadido de la interactividad y la percepción del progreso con unas puntuaciones acordes a la habilidad de cada jugador. De esta forma surgen nuestros ejercicios interactivos, que más adelante expondremos detalladamente.

Papel de los juegos en la educación

Los estudios sobre los efectos de los videojuegos abarcan las tres grandes dimensiones de la psicología humana: la afectiva (sentimientos que despiertan), la conativa (la agresividad o la impulsividad) y la cognitiva. En este último campo, la idea común entre los investigadores es la de asociar determinados tipos de videojuegos al fomento de algunas de las capacidades mencionadas. Son muchos los autores que alaban las posibilidades educativas de estos juegos (e.g. “Los Sims”, simulador social, o “Civilization”, estrategia y simulación histórica). Se decantan por esta afirmación los hermanos Le Diberdier (1998), que consideran que estos juegos serán reconocidos y utilizados en un futuro próximo como grandes instrumentos de conocimiento.

Muchos videojuegos favorecen el desarrollo de determinadas habilidades: la atención, la concentración espacial, la resolución de problemas, la creatividad, etc. Por ello, en su conjunto, los videojuegos suponen una ayuda para el desarrollo intelectual (Mandinacht, 1987; White, 1984; Okagaki y Frensch, 1994). Aquellos que juegan se pueden beneficiar de unas mejores estrategias de conocimiento, de una práctica en los modos de resolver problemas.

Entre los españoles defensores de las posibilidades educativas de los videojuegos se encuentra un grupo encabezado por Estallo (1994, 1995) y continuado por Bartolomé (1998), Calvo (2000), Gros (1997, 2000), Etxebarria (1998) y Marqués (2000). Referencia especial debemos hacer del trabajo de Estallo, por ser un pionero en las investigaciones sobre los efectos del videojuego de consola. Según Estallo, los videojuegos pueden contribuir al desarrollo tanto emocional como intelectual de los adolescentes, y llega a afirmar que “los jugadores de videojuegos suelen ser sujetos de mayor nivel intelectual que sus compañeros no jugadores”.

Otros autores españoles se refieren a otros temas relacionados con los videojuegos. Etxebarria (1998) examina las posibilidades educativas de los videojuegos comparándolos con la teoría del aprendizaje social, y se pregunta por el motivo de la atracción de los videojuegos, algo que no logran las materias escolares. Lo que sí señala es otro aspecto interesante: su potencial tratamiento en problemas de aprendizaje, al constituir un entrenamiento eficaz en aspectos psicomotores y de razonamiento deductivo. Marqués (2000) asocia cada tipo de juego con una serie de habilidades y capacidades de desarrollo de interés para el aprendizaje y la educación: los

arcade (plataformas, luchas...) contribuyen al desarrollo psicomotor y de la orientación espacial; los deportivos, a la coordinación psicomotora; aventura, estrategia y rol, a la reflexión sobre sus valores; los simuladores, al funcionamiento de máquinas; y, por último, como el caso que nos ocupa, los puzzles y de preguntas, al razonamiento y a la lógica.

Entre los aspectos positivos de aprendizaje se señalan las siguientes características: la motivación, el aprendizaje de contenidos y tareas, los procedimientos y destrezas manuales/organizativas, y las actitudes como la toma de decisiones y la cooperación. Por otro lado, al ser parte de los recursos tecnológicos actuales, ofrecen grandes posibilidades de uso en el entorno escolar.

En cuanto a los videojuegos propiamente educativos, sorprende descubrir una actitud generalizada de desconfianza por parte de los investigadores. La razón fundamental parece estar en que se consideran poco apropiados para su inserción en el medio escolar por carecer de lo que los hace más atractivos para los niños o adolescentes. Afortunadamente, la reticencia a estos juegos va decreciendo; entre otras cosas, porque cada vez resultan más estimulantes y de formatos más parecidos a los de entretenimiento normal. Un ejemplo de ello puede constatarse en la serie Touch! Generations de Nintendo para su consola Nintendo DS, con récords de ventas y aplicaciones de aprendizaje muy diversas. El más exitoso sin duda alguna, Brain Training, cuyas bondades pasamos a describir a continuación.

Inspiración en Brain Training

Con la llegada del videojuego Brain Training, un desarrollo de Nintendo Japón para su plataforma portátil Nintendo DS, se creó un nuevo género en el terreno de los videojuegos educativos, que Nintendo ha albergado bajo el título de "Touch! Generations".

Touch! Generations es una serie de innovadoras aplicaciones para Nintendo DS y para la consola de sobremesa Nintendo Wii que ofrecen nuevas y accesibles posibilidades de entretenimiento y diversión para públicos de todas las edades. Existen diversos títulos incluidos en la serie Touch! Generations, que se pueden reconocer por el logotipo situado en la esquina superior izquierda de la portada de cada uno.



Figura 3.1 Logotipo que identifica a los videojuegos de Touch! Generations

Como abanderado de este género se encuentra Brain Training, un videojuego educativo enfocado a ejercitar y entrenar la agilidad de nuestra mente. La primera vez que activemos Brain Training, se nos preguntará por nuestra fecha de nacimiento. A continuación se realizará un rápido test para determinar nuestra edad mental, metáfora sobre la que toda la aplicación girará en torno.

A partir de aquí sólo resta comenzar el entrenamiento para mejorar nuestra edad mental, a partir de más de una docena de ejercicios verbales y numéricos que habrá que realizar unos pocos minutos cada día. El calendario incorporado en el programa mantiene un seguimiento de nuestra evolución, y cada día que completemos los ejercicios recibiremos un sello como prueba.

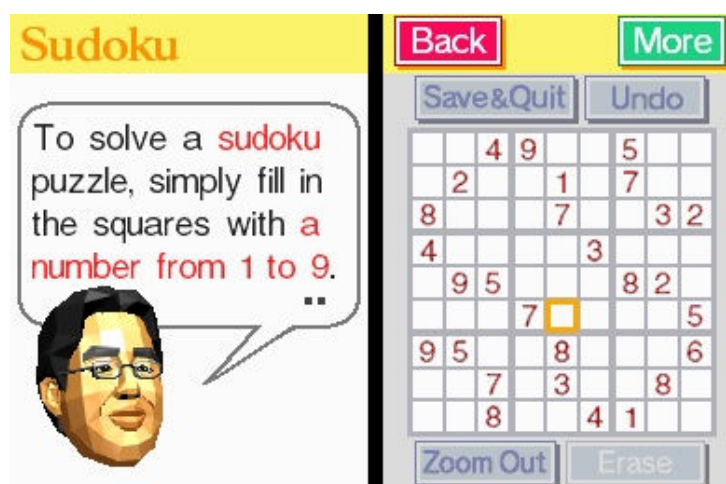


Figura 3.2 Captura de pantalla de Brain Training, con el ejemplo de un Sudoku

El gancho de Brain Training se encuentra en la Práctica Diaria, desde donde se podrá realizar un test de edad cerebral con pruebas escogidas al azar, y, sobre todo, el Entrenamiento. Se empezará con muy pocas pruebas disponibles, pero según se juegue se irán desbloqueando nuevas y, también, los modos de mayor dificultad de algunas.

Su utilidad fundamental es la que, desde hace varios años, los psiquiatras recomiendan: ejercitar la mente para evitar su envejecimiento prematuro. Brain Training es un juego que tiene esta idea por objetivo, ya que ha contado con la colaboración de un neurólogo de prestigio, Ryuta Kawashima. El Dr. Kawashima adaptó de su libro "Entrena tu cerebro: un cerebro mejor en 60 días" los mismos principios que rigen el videojuego, obteniendo un gran éxito dada la mayor interactividad que supone.

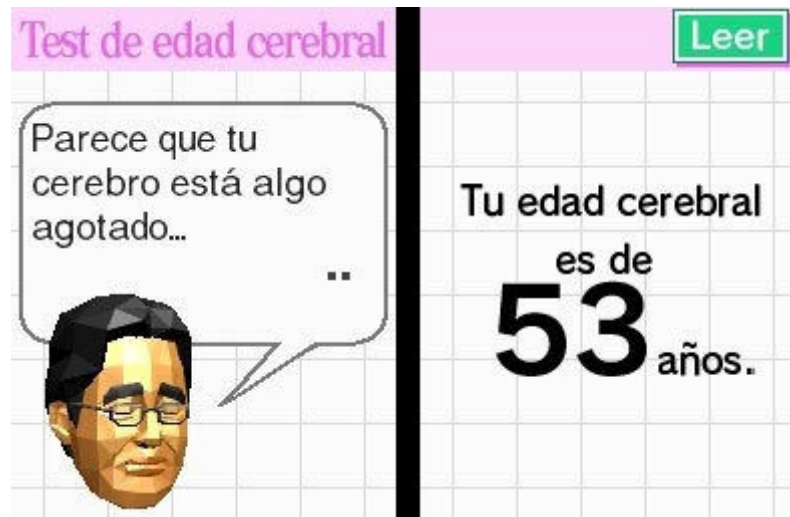


Figura 3.3 La metáfora oculta bajo Brain Training: la edad mental

A continuación se muestra en qué consisten algunas de las pruebas que se incluyen en el juego:

- **Cálculo 20:** Es una serie de 20 operaciones aritméticas sencillas, ya que la resolución de ejercicios simples produce más estimulación que la de ejercicios complejos. Escribimos el resultado en la pantalla táctil, y el objetivo es hacerlo en el menor tiempo posible, con una penalización por cada error.
- **Temporizador:** Nos aparecerán en pantalla dos relojes analógicos, marcando cada uno una hora. Nuestro objetivo es calcular la diferencia horaria entre uno y otro
- **Sonidos:** Se nos presentarán unas operaciones matemáticas sencillas, y tendremos que decir el resultado en voz alta.
- **Sudoku:** Resolución de Sudokus.

Al mismo tiempo, otras compañías han imitado a Nintendo con aplicaciones similares, ya que todo lo que implique aprender jugando o de una forma diferente cuenta con un fuerte atractivo.

En este proyecto se ha seguido la misma línea, pero queriendo ir más allá, creando casi un nuevo género con la integración de un LMS, pero reutilizando las ideas de Touch! Generations y algunas de las características de Brain Training. Si en Brain Training se medía la edad mental, en CS-Training se medirán los conocimientos sobre un tipo específico de materia de informática. Sobre el resto de características, a continuación se tabula una comparación entre Brain Training y lo que se busca en este proyecto, más allá de las similitudes entre los títulos de ambos:

Brain Training	CS Training
Se establece un régimen especial para cada usuario.	Cada alumno tiene su perfil de ejercicios y exámenes
Usa un calendario incorporado para seguir y memorizar tu progreso.	Cada vez que un alumno realiza una examen, queda registrado en una base de datos.
Obtén un cierto número de puntos para desbloquear tests y ejercicios adicionales	A medida que se van superando niveles de dificultad se pasa al siguiente curso
Se muestran los resultados en gráficos fáciles de comprender.	Se pueden usar gráficas de progreso con los datos almacenados en una base de datos
Tiene un sistema de reconocimiento de voz: puedes responder algunas preguntas con tu voz	No compatible
También incluye los números del famoso juego sudoku	Se puede incluir cualquier tipo de juego o ejercicio
Juega con tus amigos a través de la comunicación inalámbrica de Nintendo DS. Sólo se necesita una tarjeta DS y pueden unirse hasta 16 jugadores	Mediante un ranking, se puede comparar en qué puesto está el alumno, además de poder hacer grupos para la ejecución de ejercicios de colaboración
Envía una versión demo gratis a un amigo mediante las descargas inalámbricas	Sería una aplicación de código abierto, a la cual cualquier persona tendría acceso

Tabla 3.1 Comparación de características entre Brain Training y CS Training

Como hemos citado anteriormente, queremos potenciar la rivalidad y competitividad entre los alumnos al usar la aplicación, y para ello, hemos de dejar constancia de los resultados que estos obtengan cuando realicen sus ejercicios. La importancia de que estos resultados sean visibles de una forma organizada, relacionados con la materia a la que se refieran, e identificativos del progreso que el alumno consigue, es el principal motivo por el cual existe una necesidad de almacenar y manejar esta información. Gracias a los Sistemas de Gestión del Aprendizaje, conocidos como LMS, podemos dar cabida y convertir en un hecho todas estas ideas.

e-Learning y LMS

El e-learning, o aprendizaje electrónico, constituye una propuesta de formación que contempla su dominio en internet, haciendo uso de los servicios y herramientas que esta tecnología provee, ya sea vía web, aplicaciones locales o por correo electrónico.

Es una de las opciones que actualmente se utiliza con mayor frecuencia para atender la necesidad de educación continua o permanente. El e-learning se expande a los jóvenes que pueden empezar una carrera universitaria o algún tipo de enseñanza y tiene la ventaja de que los usuarios eligen sus propios horarios, y pueden entrar a la plataforma desde cualquier lugar donde puedan acceder a una computadora y tengan conexión a internet.

Con el uso de una aplicación software y un LMS, se pretende utilizar el e-learning no sólo en su modalidad a distancia, sino como apoyo a las clases presenciales de un modo mixto, algo que se conoce como B-learning o Blended Learning. La modalidad B-Learning contempla el desarrollo conjunto de clases on-line y presenciales, digamos semi-presenciales. De este modo, los alumnos están presentes en cada lección realizando sus trabajos en la aplicación y comunicándose con el LMS en el propio aula. Así, el profesor asigna tareas y comprueba su realización sin necesidad de llevar un seguimiento personalizado por cada alumno, sino de una manera global y con mayores facilidades.

Learning Management Systems

Un Learning Management System (a partir de ahora, LMS) es el elemento central de la comunicación en e-learning. Se trata de una aplicación, generalmente basada en la web (y en nuestro caso, de código abierto), en la que profesores y alumnos pueden implementar sus materias y cursarlas, respectivamente, de forma virtual. Esto se consigue creando cursos virtuales en los que se alojarán desde apuntes y ejercicios hasta tests calificables, y en los que los alumnos dados de alta podrán, gracias a su cuenta de usuario, comentar sus dudas en foros, ver publicadas sus calificaciones y las de los demás, realizar tareas, etc. Son sistemas encargados de la gestión y administración de los recursos educativos que ponen al alcance de quienes estén interesados.

Integrada junto a un videojuego de corte educativo, como es el caso, este tipo de herramienta permite la comunicación de resultados independientes para cada jugador, una organización de sus propios tests atendiendo a las necesidades del curso en los que se realicen, etc. En definitiva, supone una ampliación lógica y necesaria al modelo de aplicación de Brain Training seguido en este proyecto, si se quiere enfocar de una manera más directa a la enseñanza tradicional y al aprendizaje escolar.

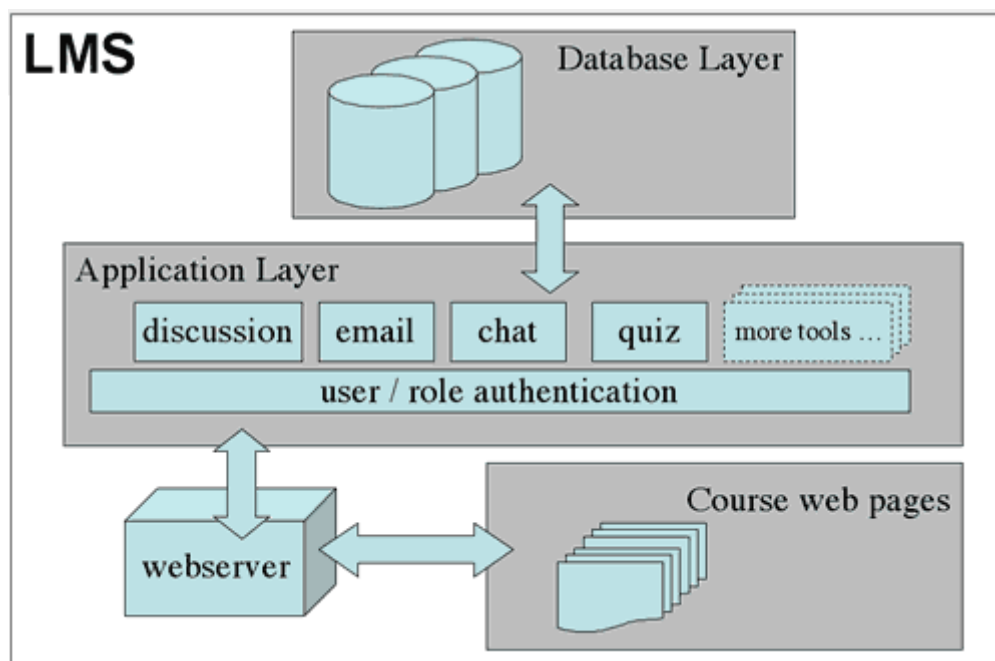


Figura 3.4 Esquema de organización de un LMS genérico

Con el uso de estas plataformas no todo son ventajas, encontramos ciertos inconvenientes debido a que se trata de sistemas abiertos y heterogéneos (e.g. distintos ordenadores con sistemas operativos y LMS distintos) y difieren en las herramientas disponibles o en aspectos como la comunicación entre distintos sistemas. Se dificulta por tanto la reutilización o adaptación de contenidos y aplicaciones cuando la plataforma (o, incluso, el contexto educativo) cambia. De cara a paliar o minimizar este problema, surge un modelo para el diseño de cursos denominado modelo de objetos de aprendizaje (OA) u objetos educativos (*learning objects*).

Wiley (2000) define estos objetos educativos como: “cualquier recurso digital que pueda ser reutilizado como soporte para el aprendizaje”, matizando su uso para designar material educativo diseñado de cara a maximizar las veces en las que se puede utilizar dicho recurso. Esta idea se recoge también en la definición de Polsani (2003): “unidad didáctica de contenido, autocontenida e independiente, predispuesta para su reutilización en múltiples contextos instruccionales”. Podemos comprobar la importancia de diseñar los contenidos de una manera casi automática y estándar, para que sean reutilizados por distintos sistemas, cobrando gran protagonismo las herramientas de autoría para estos recursos.

Durante el desarrollo de CS-Training se ha tenido en consideración el modelo de objetos educativos, proporcionando un modelo conceptual estándar para los ejercicios interactivos (nuestros recursos educativos) contenidos, y apoyando esa idea con una comunicación entre la plataforma Nintendo DS y Moodle abierta a ordenadores personales distintos con sistemas operativos distintos. La elaboración de una herramienta de autoría sería un paso lógico y directo a partir del desarrollo actual.

Desarrollo de este documento

Tras haber expuesto la motivación y haber definido los beneficios del e-Learning y la integración con un LMS, lo que sigue en este documento trata de plasmar detalladamente el trabajo de un año de investigación y desarrollo.

En primer lugar se contempla el modelo con el que se pretende describir de forma general la aplicación y su conjunto de ‘mini-juegos’ o ejercicios interactivos, los diferentes modos de uso incluyendo la comunicación con un LMS y los posibles campos de aplicación.

Se continúa con la descripción completa de CS-Training, como ha sido denominado este desarrollo, los papeles que la plataforma Nintendo DS y el LMS desempeñan, y cada tipo de ejercicio interactivo con una instancia del mismo.

Seguidamente, se informa sobre la arquitectura general del proyecto, en una visión de alto nivel, con diagramas concretos.

A continuación, se detalla el proceso de desarrollo seguido en el curso, dividido en tres iteraciones, las cuales se subdividen en micro-iteraciones, con una descripción del trabajo realizado y los problemas y riesgos tecnológicos encontrados.

Finalmente, llegamos a las conclusiones del trabajo y el alcance del mismo. Añadimos las referencias utilizadas para la investigación y desarrollo en este proyecto y un par de apéndices, en los que se describen las librerías de PALib para el desarrollo en NDS, la comunicación realizada con Moodle y los manuales del profesor y el alumno.

4. Modelo General



Figura 4.1 Esquema del modelo general de la plataforma CS Training

El desafío tecnológico que implica el modelo de CS-Training consiste en la necesidad de establecer una comunicación e integración entre los servicios de un LMS como es Moodle y la plataforma portátil Nintendo DS. Gran cantidad de aplicaciones de software de aprendizaje electrónico ejecutadas en un PC incorporan la característica de una integración total con Moodle, siendo un claro ejemplo de ello el proyecto del grupo de investigación e-UCM, eAdventure, pero la misma idea es innovadora en el ámbito de las videoconsolas.

Por un lado, el LMS es el encargado de administrar y gestionar los cursos, alumnos y lecciones de los que se compondrá la vertiente más cercana al modelo de enseñanza tradicional de este proyecto. Por otro, la videoconsola contendrá la aplicación educativa con la que el alumno interactuará.

De esta forma, la comunicación se establece en dos vías bien diferenciadas:

- **Moodle -> NDS:** Esta primera vía de comunicación permite al LMS configurar determinadas facetas de la ejecución de la aplicación, como puede ser la dificultad del test a realizar.
- **NDS -> Moodle:** La aplicación comunicará al LMS los resultados del alumno a cada ejecución, además de otros parámetros relativos a esta.

Ejercicios Interactivos o MiniJuegos

Como hemos visto, el modelo conceptual de CS Training está basado plenamente en la interacción y comunicación entre la plataforma *Nintendo DS* y el Sistema de Gestión del Aprendizaje *Moodle*. Dentro del ámbito de la videoconsola, la aplicación incluye una serie de

ejercicios interactivos evaluables encapsulados bajo un cargador y evaluador global, encargado de procesar las comunicaciones entre la aplicación y el LMS. Cada ejercicio interactivo sigue un determinado patrón conceptual, atendiendo a un modelo general que se define a continuación.

Partiendo de la base de un ejercicio teórico en una determinada asignatura, se añade el toque de interactividad que supone el uso del Stylus (lápiz que se usa en la pantalla táctil de la plataforma Nintendo DS) y el conjunto gráfico de animaciones que genera la videoconsola, lo que enriquece la experiencia de resolverlo. Además, elementos como un temporizador, que impone un límite estricto al tiempo disponible, y el hecho de obtener una calificación como resultado de la realización, motivan la parte competitiva y desafiante del juego.

Siguiendo los parámetros habituales en el subgénero de Touch Generations!, hasta 10 tipos de minijuegos (o ejercicios interactivos) conforman la parte práctica y educativa de CS Training. Estos ejercicios exploran todo tipo de conocimientos impartidos en las aulas de la Facultad de Informática de la UCM, desde el primero hasta cuarto curso. Se trata de tests prácticos basados en ejercicios reales que se resuelven en clase, a los que se añade el toque de la interactividad, la percepción de jugabilidad, una vistosidad inherente a los juegos electrónicos, etc. Estos minijuegos responden a unos modos de uso específicos según el modelo general.

Modos de uso



Figura 4.2 Esquema de los modos de uso principales de CS Training

Atendiendo al papel que el profesor desempeña en este modelo, la aplicación distingue los siguientes modos de uso:

Práctica Libre o "Entrenamiento" (no supervisado)

En este modo de práctica libre el alumno puede ejercitar tantas veces como desee, sin presión por obtener una determinada calificación o el apremiante tiempo, las distintas pruebas de que dispone CS Training. Basta con elegir la rama, ya sea *hardware* o *software* y, seguidamente, el ejercicio interactivo en cuestión. De esta manera se permite la mejora significativa de la destreza y

habilidad en la resolución de los problemas gracias a una práctica continuada de un determinado formato de ejercicio. Este modo de juego no está supervisado, por lo que los resultados obtenidos serán única y exclusivamente para el propio alumno, que comprobará por sí mismo su nivel de conocimiento.

El fundamento de disponer de un modo de práctica responde a la misma idea que la aplicación, y no es otro que la posibilidad de afianzar conceptos y coger soltura en la resolución de los problemas que igualmente se encontrarán durante las clases prácticas de cada asignatura, de un modo más dinámico e interactivo.

Test Evaluados o "Evaluación" (supervisado)

La modalidad de evaluación difiere en gran medida con lo que se propone en la práctica libre: ahora las acciones del alumno sí se verán reflejadas en una calificación de relevancia. Los tests que se habilitarán en este modo se generan a partir de numerosas instancias de los ejercicios interactivos, cada una de las cuales comunicará su nota a un evaluador global, que publicará una ponderación como resultado final en el sistema de gestión del aprendizaje dispuesto al efecto. De esta forma, las respuestas de los alumnos están siendo supervisadas por el instructor, que dispondrá de un seguimiento completo gracias a los servicios que prestan los LMS, como un historial de intentos, o una valoración global del conjunto de realizaciones.

Estos tests evaluados pueden configurarse desde la plataforma de gestión o LMS, siguiendo una rutina específica, eligiendo desde su aleatoriedad hasta dificultad y número de instancias incluidas. Será función del instructor configurar sus características de acuerdo a sus intenciones en cada curso.

De las dos modalidades posibles, la evaluación sugiere la necesidad de la integración de un instructor o profesor en el ámbito de la aplicación, además de confirmar la existencia de un sistema de puntuación que permite un seguimiento de la evolución constante del alumno.

Participación del profesor

Frente a otras aplicaciones de e-learning, CS Training integra en primera instancia a la figura del instructor. Las funciones que debe desempeñar se centran en la gestión de las herramientas educativas a su disposición en Moodle, además de una administración de los alumnos a su cargo en los distintos cursos o lecciones que imparta. Así, el profesor es el encargado de configurar los tests evaluados que los alumnos ejecutarán en la videoconsola cuando establezcan comunicación con el LMS.



Figura 4.3 Roles y funciones del profesor en CS Training

Las labores de seguimiento y apoyo a la docencia que ofrece Moodle no suplen la necesidad de que un instructor lleve a cabo sus funciones, y así queda reflejado en el modelo que fomenta CS Training, donde los profesores se verán en la obligación de generar todo el contenido educativo que los alumnos realizarán, así como adaptar este contenido a las necesidades de cada alumno en concreto.

Puntuación evolutiva

Por cada intento que el alumno realiza en los tests evaluados se genera una calificación que se almacena en su historial de ejecuciones dentro de Moodle. Este historial permite llevar a cabo una monitorización clara del progreso del alumno, así como establecer un ranking con las puntuaciones más altas para cada alumno que haya realizado el test. Dicho ranking permanecerá a la vista de todos los usuarios del LMS, fomentando la rivalidad y competencia por conseguir las puntuaciones más altas.

Posibles campos de aplicación

Como se ha podido comprobar, el modelo de CS Training es altamente escalable de cara a tratarlo con un sinfín de materias específicas. Ya sean estudios de grado superior, carreras universitarias, formaciones especiales en áreas técnicas, todas tienen cabida en un modelo de estas características. Diferentes ejercicios para cada materia o apartado, que en conjunto formarán tests evaluables, y bajo el soporte de un sistema de gestión del aprendizaje con seguimiento de un instructor para todos los alumnos en el curso. Así, pasamos a enumerar posibles aplicaciones de este modelo en otras ramas o áreas de conocimiento, que perfectamente podrían haber sido un desarrollo de este proyecto de investigación:

- **Autoescuela Virtual:** Conocimientos teóricos y algunos prácticos sobre cómo conducir un vehículo. Asimilación de conceptos referentes a límites de velocidad,

señales de tráfico, todo bajo un atractivo aspecto gráfico que represente con fidelidad las sensaciones que implican conducir un vehículo.

- **Curso de Botánica:** El aprendizaje de las plantas más comunes y también de las más exóticas. Nos ayuda a saber cómo cuidarlas y cuál sería su mejor entorno.
- **Físicas:** Ecuaciones básicas, leyes de los principales físicos llevadas a la práctica. Ejercicios sobre cinética, electromagnetismo, con la interactividad que añade nuestro modelo conceptual.
- **Matemáticas Aplicadas:** Aprendizaje y aplicación de teoremas, ecuaciones, cálculo con matrices y representaciones gráficas de funciones.
- **Fisionomía:** Conocimiento del cuerpo del ser humano y su estructura, con ejemplos visuales significativos.

5. CS – Training

Computer Science Training (de ahora en adelante, CS-Training) es el innovador servicio que se ha creado para la interacción entre la plataforma Nintendo DS y el LMS Moodle. En este apartado se hará una descripción de la infraestructura que llega a ser CS-Training, los papeles fundamentales que juegan la Nintendo DS y Moodle en la misma y la descripción de los que son nuestros objetos educativos, los ejercicios interactivos, y sus tipos, junto con las instancias específicas que se han realizado de ellos.

Descripción General

CS-Training es una infraestructura en la que juegan un papel muy importante tanto los dispositivos de videojuegos portátiles y los LMS para el aprendizaje y la práctica de conocimientos de la carrera de Ingeniería Informática (en el caso que nos ocupa, la impartida por la Facultad de Informática de la UCM).

Se busca como propósito principal que sea un apoyo a las clases asistenciales, donde el alumno pueda conseguir puntos adicionales al realizar de forma correcta los ejercicios propuestos en el modo evaluación que se describe más adelante, y además pueda practicar libremente para adquirir nuevos conocimientos y mejorar su destreza.

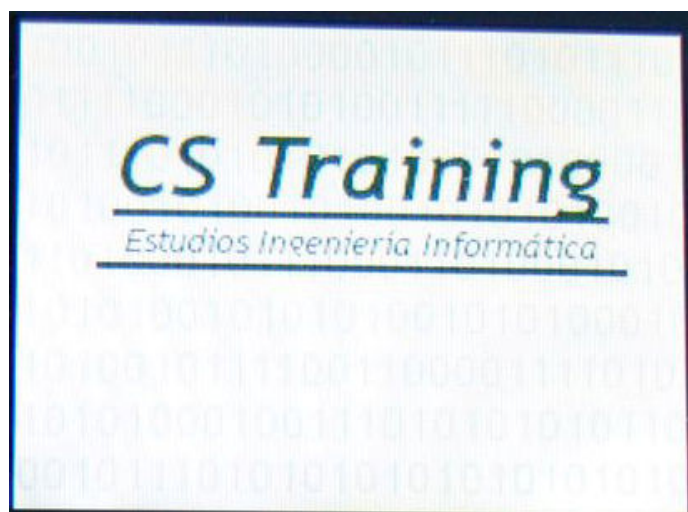


Figura 5.1 Captura de la pantalla de presentación de CS Training

Distinguiremos entre dos ámbitos principales en CS-Training:

Ámbito Nintendo DS

En el ámbito de la Nintendo DS, CS-Training como videojuego educativo pertenece a un nuevo subgénero, derivado de la familia de Brain Training, Touch Generations!, e ideado en este trabajo.

Posee dos modos de uso, uno de entrenamiento o práctica libre y otro de evaluación o de tests evaluados. En el primero, el alumno practica los diferentes tipos de instancias de ejercicios interactivos, según el perfil de dificultad que elija, algo íntegramente relacionado con el curso y las asignaturas que tenga matriculadas en la Facultad. El segundo modo propone al alumno un conjunto de ejercicios, por defecto no relacionados, y establecidos por el profesor.



Figura 5.2 Captura del menú principal de CS Training

La mayor parte de la lógica de la aplicación se encuentra en la Nintendo DS. Por consiguiente, todos los módulos de evaluación, de interacción y configuración están incluidos en el cartucho de la videoconsola portátil, a excepción de la configuración de los ejercicios de las evaluaciones, que se reciben desde el LMS Moodle.

La aplicación se estructura como un cargador de instancias de ejercicios interactivos, los cuales se tratan como módulos. La forma de crear esos módulos es estructuralmente sencilla, aunque, dependiendo del tipo de ejercicio, se puede complicar la propia lógica del juego. Cada ejercicio interactivo tiene un evaluador local, donde se comprueban los aciertos o fallos que se han cometido en su ejecución. La medida generada por estos evaluadores locales no es estándar, por lo tanto, es necesaria la integración de un evaluador global encargado de normalizar el conjunto de evaluaciones para su correcta comprensión.

Ámbito LMS – Moodle

Moodle juega un papel muy importante a la hora de comunicar al alumno y al profesor en el sentido del progreso de la realización de los test. En él quedan establecidos los perfiles de cada usuario, en los que figurarán los cursos matriculados, los tipos de test a los que tiene acceso, etc. Toda la labor administrativa y de gestión requiere la existencia de un administrador general, rol que podría desempeñar o bien el instructor (con mayor número de tareas globales) o bien una persona ajena a la aplicación (con mayor independencia).

Cada curso creado por el instructor en Moodle estará formado por el conjunto de tests a los que los alumnos pueden acceder. Estos tests representan conceptualmente el grupo de ejercicios interactivos que se realizarán en un futuro en la aplicación de la videoconsola, y guardan un registro de intentos por cada alumno matriculado en el curso. Dichos intentos contendrán la calificación final junto con el tiempo requerido en su solución.

El conjunto de intentos realizados se representa en un historial global, que juega dos papeles muy importantes. Por un lado, el instructor podrá observar los progresos del alumno en el curso y por otro, todos los alumnos estarán ordenados por un ranking de mejores puntuaciones. Esto último motivará y fomentará una sana rivalidad entre los alumnos por obtener las mejores puntuaciones posibles, progresando indirectamente en su nivel de conocimientos sobre las materias.

Ejercicios Interactivos (E.I.)

El concepto de Ejercicio Interactivo ha sido adoptado por descripción propia de los sub-productos. Hasta diez tipos distintos de instancias de ejercicios se pueden encontrar en CS-Training, ya sean referentes al campo de software como al campo de hardware. La interacción en ellos se realiza por medio del Stylus, el lápiz táctil de Nintendo DS, dando un valor a las entradas. La aplicación nos proporcionará las salidas dependientes de cómo de correctas o incorrectas sean esas entradas.

La estructura de todo ejercicio interactivo como objeto educativo tiene dos partes:

- La pantalla principal o superior, que suele ser descriptiva e informativa para el alumno sobre algún evento o señal relevante del ejercicio.
- La pantalla inferior, donde se lleva a cabo la mayor parte de la interacción con el alumno.

Con más detalle, en la interacción alumno-aplicación pueden describirse los siguientes tipos de elementos:

Elementos de interacción

- **Piezas:** en los ejercicios de arrastrar elementos de contenido, las piezas son importantes porque según su colocación o el tratamiento que se les dé por parte del alumno, dependerá la nota obtenida en el ejercicio interactivo. El ejemplo que se muestra es de una pieza del ejercicio interactivo ‘Herencia y Polimorfismo’



- **Botones:** Los botones activan o facilitan el acceso a las diferentes partes de la aplicación. En el ejemplo, se muestra uno de los botones para acceder al ejercicio interactivo de ‘Visibilidad de variables’.



- **Nodos de árboles:** Se pueden mover y/o pinchar sobre ellos para interactuar. Aparecen en los ejercicios de software que tienen que ver con estructuras arbóreas o en ejercicios de hardware para recorrido de distintos tipos de circuitos.



- **Flechas:** Indican acciones como avanzar, retroceder o, incluso, desplazar el fondo de una instancia de un ejercicio. Su mayor aparición es en los menús de CS-Training para Nintendo DS.



Elementos de control

- **Botones físicos:** Respecto a los controles en la aplicación, la figura siguiente muestra cómo usar CS Training para DS con los botones de la propia consola.



Figura 5.3 Captura de pantalla de la configuración de los controles

- **Botones software:** Estos botones se encuentran en la aplicación software y proporcionan acceso a otras partes de la aplicación o bien ejecutan algún tipo de evento. A continuación, se muestran algunas figuras de botones en la aplicación

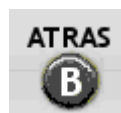
Pulsando con el Stylus sobre este botón se corrige el ejercicio



Pulsando con el Stylus sobre este botón se rehace o se crea un nuevo ejercicio



Para salir del ejercicio hay que dar a este botón



Elementos de contenido

- **Enunciados:** Los enunciados describen el objetivo del ejercicio y alguna que otra indicación para realizarlo.

Enunciado de la instancia del ejercicio de 'Puzle Piezas'

ENUNCIADO:
 Completa con las piezas correctas un programa que complete 10 iteraciones y en cada una muestre por pantalla el número de iteración actual.
 Pincha sobre las piezas y arrástralas a la posición que creas adecuada.

- **Temporizadores:** Muestran el tiempo del que dispone el alumno para acabar el ejercicio que este realizado. Cuando el temporizador llegue a cero, el ejercicio automáticamente se corregirá con lo que estuviera hecho. El tiempo estándar que se le asigna a los ejercicios interactivos es de cinco minutos.

El tiempo es importante al hacer un ejercicio

... cuando tocando el
 ... pendiente.
 3:37

- **Notas:** Muestran el resultado de un ejercicio al completarlo dando a corregir. Si el alumno pasa el ejercicio se le indicará con letras verdes la calificación obtenida, si fuera así se indicará con letras rojas y la nota obtenida.

Corrección del ejercicio

... nodo correspondi
 Bien hecho!!
 Nota: 10

A su vez, en la resolución de cada uno de los ejercicios interactivos se distinguen unas ciertas fases, algo que podría asemejarse a las pasos habituales que se realizan en las clases prácticas de las asignaturas reales:

Primera Fase o Fase de Lectura y Comprensión: En cada ejercicio interactivo se realiza una cuenta atrás, de tres a uno, para que haya el tiempo suficiente a leer y comprender lo que se pide en cada ejercicio. De cualquier forma, durante el desarrollo del ejercicio interactivo, estará disponible para el alumno en todo momento.

Cuenta atrás que se realiza para la que el alumno se prepare

Quedan 3 segundos

0001010100101010001011101010
101010010100011010101001010
11110001010101000100
1100100100011110100101000
1001010111000100100011110
11010001000101010010101000
0101010110110001010010100
10101010001010100101000
11010000011100100100110101
0101010100101000101000101

3

Quedan 2 segundos

0001010100101010001011101010
101010010100011010101001010
11110001010101000100
1100100100011110100101000
1001010111000100100011110
11010001000101010010101000
0101010110111110010100
10101010001010100101000
11010000011100100100110101
0101010100101000101000101

2

Quedan 1 segundo

0001010100101010001011101010
101010010100011010101001010
11110001010101000100
1100100100010100101000
1001010111000100100011110
11010001000101010010101000
0101010110110011110010100
1010101000101010100101000
11010000011100100100110101
0101010100101000101000101

1

¡Empieza el ejercicio!

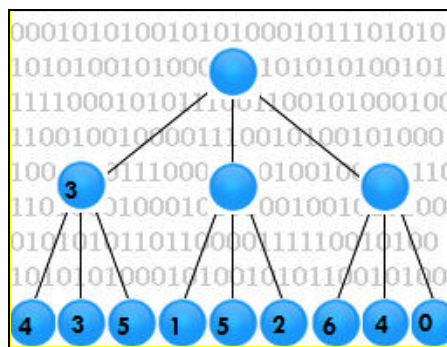
0001010100101010001011101010
101010010100011010101001010
11110001010101000100
1100100100010100101000
1001010111000100100011110
11010001000101010010101000
0101010110110011110010100
1010101000101010100101000
11010000011100100100110101
0101010100101000101000101

GO!

Enunciado que aparece en todo momento en la cuenta atrás

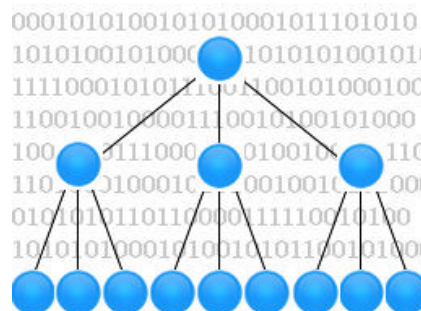
ENUNCIADO:
 Completa con las piezas correctas un programa que complete 10 iteraciones y en cada una muestre por pantalla el número de iteración actual.
 Pincha sobre las piezas y arrástralas a la posición que creas adecuada.

Segunda Fase o Fase de Realización del Ejercicio: El alumno interactúa con la pantalla inferior proporcionando la solución que cree oportuna. Para llevar a cabo la realización de los ejercicios se dan hasta cinco minutos, tiempo más que suficiente para terminar correctamente cualquier tipo que se plantee.



Tercera Fase o Fase de Corrección del Ejercicio: Al pulsar en el botón de corregir, el evaluador local de dicho ejercicio se ejecutará y proporcionará una nota a la realización del ejercicio por parte del alumno

Cuarta Fase o Fase de Reintento/Nuevo Ejercicio: En esta fase, si el alumno desea practicar más o comprender mejor la materia puede reintentar el mismo tipo de ejercicio interactivo, exclusivamente en el modo entrenamiento, ya que en el modo de evaluación no se dispondrá de esta opción. Si se elige esta opción se volverá a la fase 2.



Quinta Fase o Fase de Salida: Una vez finalizadas las anteriores fases, el alumno puede optar por salir del ejercicio pulsando el botón B de su Nintendo DS.

A continuación describiremos el conjunto de modelos de ejercicio interactivo que hemos desarrollado para CS-Training, con sus instancias particulares, y sus características como objetos educativos reutilizables.

Ejercicio Interactivo 1 - Puzzle de Programación

Descripción:

Uno de los aspectos más importantes en la informática es la programación. Por ello, en este tipo de ejercicios se refuerzan las estructuras de segmentos de código específicos sin utilizar un lenguaje de programación en concreto (pseudo-código). Abarca asignaturas como Introducción a la Programación o Laboratorio de Programación (1,2 y 3).

Los conocimientos que se adquieren, de una manera más detallada, son los siguientes:

- Sintáctica de los lenguajes de programación (general)
- Funcionamiento de los lenguajes de programación
- Sintáctica de un lenguaje de programación concreto. En la última versión no se dispone de esta modalidad, pero no se descarta hacer una expansión para tal fin.

El funcionamiento del ejercicio es completamente interactivo: se utiliza fundamentalmente el lápiz táctil (Stylus) de la consola. El usuario deberá arrastrar las piezas de un supuesto programa o segmento de programa hacia las posiciones vacías de una manera adecuada, para que dicho fragmento de código se correcto conceptualmente.

Existen tres tipos de ejercicios dentro de este modelo:

- Ejercicios de sintaxis: El objetivo principal es construir programas o partes de programas (bucles, sentencias “if”, etc.) que sean sintácticamente correctos.
- Ejercicios de funcionamiento: Se pone a prueba la capacidad para saber cómo funciona una función, bucle, etc. Se pide al jugador que responda qué resultado va a dar o qué valor debe tener una variable para que dé cierto resultado.
- Híbridos: Son los ejercicios más avanzados y son una mezcla de los dos anteriores. Se deberán construir programas completos o funciones que realicen tareas específicas.

Las reglas son las siguientes: Se nos proporcionan partes de un programa en forma de piezas de un puzzle. Se pueden usar el número de piezas que se quiera y colocarlas en cualquier orden. Algunos ejercicios tienen variables a las que se les puede asignar un valor. Una vez se esté satisfecho con la disposición de las piezas y el valor de las variables se pedirá al juego que evalúe el resultado. El número de aciertos se puntuará, así como la progresión y la curva de aprendizaje.

Percepción del Progreso y Realimentación:

Se va obteniendo según se van superando ejercicios y la dificultad de estos aumenta. Los modos de aumentar la dificultad consisten en:

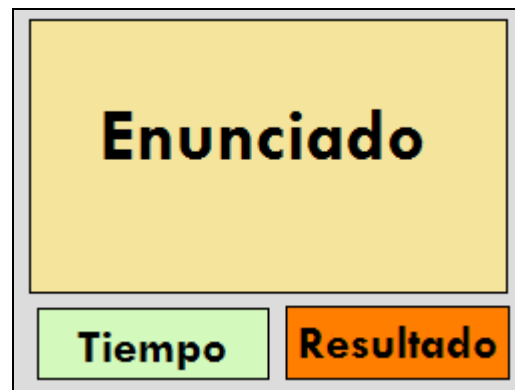
- Poner “piezas” que no corresponden, de modo que el jugador deba seleccionar también cuales usar y no sólo el orden.
- Incluir estructuras de datos complejas como, por ejemplo, punteros.
- Introducir conceptos complejos como la recursividad.

Evaluación:

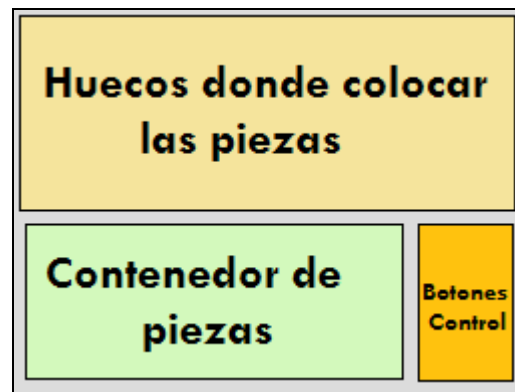
Cada pieza mal colocada en su casilla se contempla como un fallo. Una característica importante de la programación estructurada es, precisamente, su estructura, por lo que no se pueden cometer más de 2 fallos si se quiere superar satisfactoriamente el ejercicio. Dos fallos ya implican que un pseudo-código no obtenga el resultado deseado, pero al tratarse de un ejercicio de primer curso, donde los alumnos no tienen aún la soltura necesaria y están encontrándose por primera vez con este tipo de conceptos, se evalúa de una manera más benévola. En niveles superiores de dificultad se tendrán en cuenta otras consideraciones.

Generalización del ejercicio:

- **Pantalla Superior:** Consta de tres elementos de contenido, un enunciado, en el que se sitúa la función del programa que se quiere que se forme con las piezas de la pantalla inferior, el tiempo y el resultado, que se mostrará cuando se corrija el ejercicio.

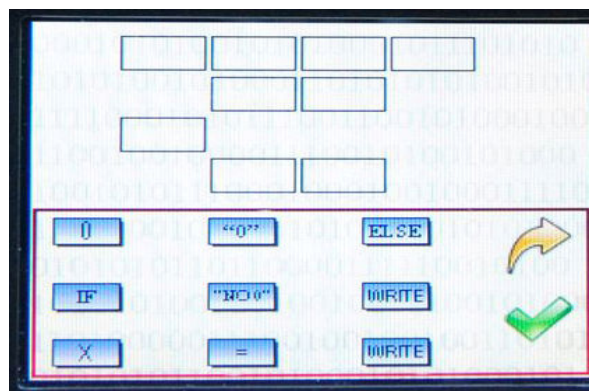


- **Pantalla Inferior:** Consta de dos contenedores principales, uno de huecos y otro donde se colocan las piezas que el alumno debe de situar correctamente en los huecos. No hay que olvidar los botones de control: rehacer y corregir.

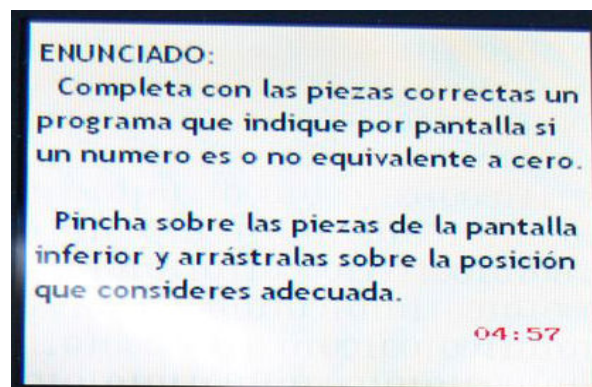


Instancia en la aplicación de este proyecto en la NDS:

- Pantalla inferior



- Pantalla Superior



Ejercicio Interactivo 2 - Recorrido de árboles

Descripción:

Los recorridos ordenados de árboles son conceptos de mucha utilidad en el paso de una estructura de datos arborescente y por lo tanto, recursiva, a un vector. Existen tres tipos fundamentales, dependiendo del orden en que se visitan los nodos del árbol: pre-orden (se visitan por orden subárbol izquierdo, raíz y subárbol derecho), in-orden (raíz, subárbol izquierdo, subárbol derecho) y post-orden (subárbol izquierdo, subárbol derecho y, por último, raíz)

La asignatura que refuerza este ejercicio interactivo es la de Estructuras de Datos y de la Información, de gran importancia en la organización y estructuración de toda la información a tratar en las diferentes aplicaciones que se desarrollen.

Los conocimientos que se adquieren, de una manera más detallada, son los siguientes:

- Estructuras de datos: Árboles binarios.
- Recorridos de árboles: in-orden, pre-orden y post-orden.

El funcionamiento básico del ejercicio consiste en seleccionar los nodos del árbol (que se quiere transformar en un vector) por medio del método de recorrido especificado en el enunciado, y arrastrarlos hasta la posición del vector que corresponda. El jugador podrá seleccionar el nodo que desee y colocarlo en una posición que considere correcta.

El objetivo final implica la deducción correcta del orden a seguir al recorrer un árbol binario mediante el tipo de recorrido que se pide. Cuando se halla recorrido el árbol en su totalidad el juego corregirá el ejercicio. En el caso de haber cometido algún error, se iluminarán los nodos que se han marcado incorrectamente y se indicará el camino correcto junto al del jugador.

Percepción de Progreso y Realimentación:

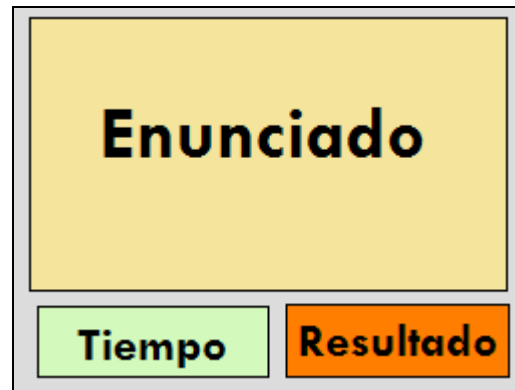
Estará representada por la complejidad de los árboles, que irá aumentando según el jugador vaya acumulando éxitos consecutivos. Se tendrá en cuenta la curva de aprendizaje: la media de éxitos al día o cada cierto número de intentos.

Evaluación:

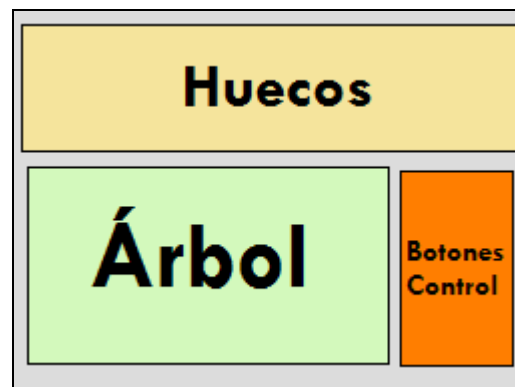
Para este tipo de ejercicio, cada nodo mal colocado en la lista de los huecos se considera un fallo al no concordar con la ordenación que le corresponde. Para que el ejercicio sea considerado superado, únicamente se pueden tener 2 fallos, es decir, dos piezas en huecos que no les corresponden, ya que no se puede permitir tener más errores en ejercicios de este tipo por ser tan básicos.

Generalización del ejercicio:

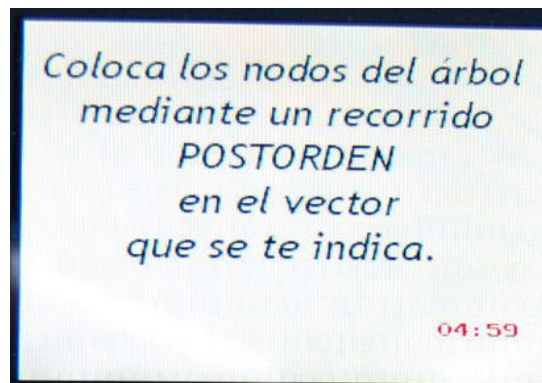
- Pantalla Superior: Se sigue el esquema de la pantalla superior del ejercicio anterior, con tres elementos de contenido: el enunciado, en el que se detallará si el alumno debe de hacer un recorrido post-orden ó pre-orden (instancias desarrolladas).



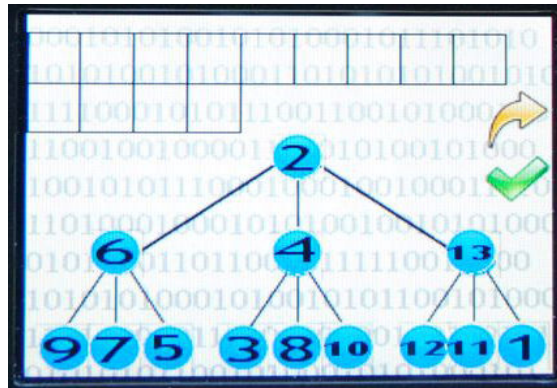
- Pantalla Inferior: Con tres elementos de contenido, un contenedor de huecos en forma de vector, otro donde se sitúa el árbol binario con sus nodos y, finalmente, los botones de control.

**Instancia en la aplicación de este proyecto en la NDS:**

- Pantalla Superior



- Pantalla Inferior



Ejercicio Interactivo 3 - Visibilidad de variables

Descripción:

Otro aspecto importante que hay que tener en cuenta a la hora de programar es la visibilidad y el alcance de las diferentes variables de cada programa, ya sean globales o locales. Dicho de otra manera, cuándo modifican su valor, dónde se pueden utilizar y según con qué propósito.

La asignatura que se refuerza en este ejercicio interactivo no está del todo definida, es decir, este tipo de conocimiento se refleja en varias asignaturas diferentes, como Introducción a la Programación, Metodología y Tecnología de la Programación e incluso en Estructuras de Datos y de la Información.

Los conocimientos obtenidos mediante la práctica de estos ejercicios son:

- Ámbito de las variables en los lenguajes de programación.
- Uso específico de las variables en cada fragmento.

El objetivo del ejercicio consiste en obtener la salida por pantalla adecuada de un programa dado, por orden de ejecución.

Como enunciado del problema se presentarán las líneas de código de un programa determinado, con sus variables. En el área de interacción se mostrarán ciertas ranuras en orden que se corresponden con el número de salidas por pantalla del programa. A partir del código, el jugador deberá conocer el ámbito de sus variables y dejar constancia de ello, colocando el valor que se mostrará por pantalla para una determinada variable en la ranura adecuada, y de forma ordenada. El juego evaluará si la respuesta es correcta e indicará la solución en caso de fallo.

Percepción de progreso y realimentación:

Los ejercicios se complicarán conforme el alumno vaya acumulando respuestas correctas. Se evaluará la curva de aprendizaje. Para aumentar la dificultad se pueden anidar varias funciones

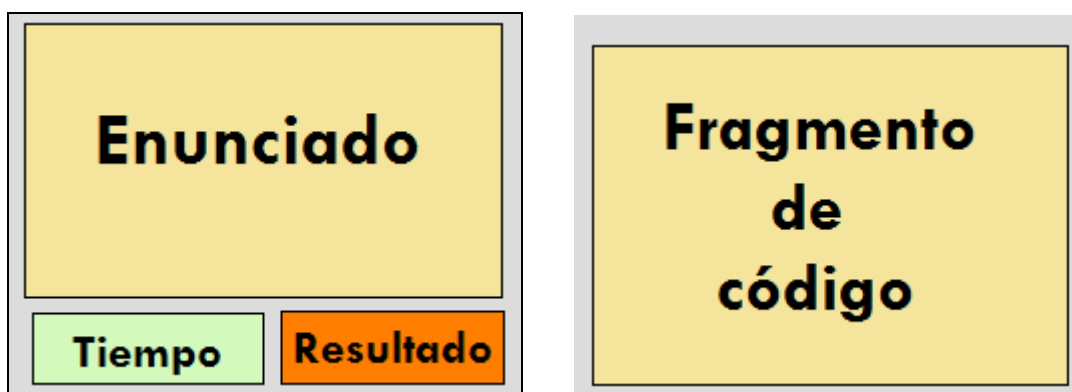
y/o poner en distintos ámbitos variables con nombre compartidos y que dichas funciones sobrescriban el valor de éstas.

Evaluación:

En este tipo de juego, un fallo se contabiliza cuando el usuario ha colocado mal alguna pieza que representa a una salida del programa propuesto. Con dos fallos no es posible superar el ejercicio, ya que este campo de la informática es muy importante a la hora de hacer aplicaciones de software.

Generalización del ejercicio:

- Pantallas Superiores: En este tipo de ejercicio existen 2 tipos de pantallas superiores, que se podrán alternar pulsando el Botón A. La primera es la que se ha venido utilizando hasta ahora, con tres elementos de contenido, un enunciado, el tiempo y el resultado. Ahora la segunda pantalla constará de un fragmento de código sobre el que el alumno tendrá que basarse para completar el ejercicio.



- Pantalla Inferior: De igual manera que en ejercicios anteriores, existen 3 elementos principales de interacción, huecos, piezas y botones de control.



Instancia en la aplicación de este proyecto en la NDS:

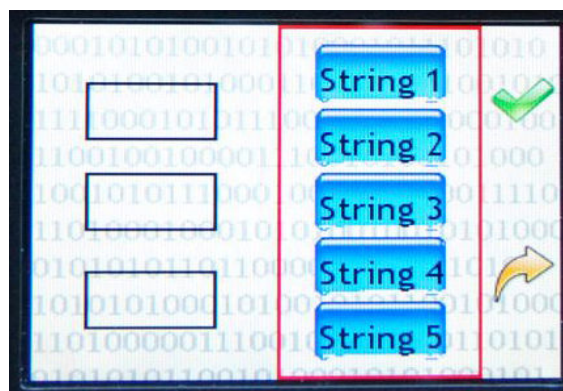
- Pantalla Superior

```

var cadena = 'String 1'
function f (){
  cadena = 'String 2';}
funcion ff (){
  var cadena;
  cadena = 'String 3';
  printf(cadena);}
int main(int arg, char * args){
  ff();
  printf(cadena);
  cadena = 'String 4'
  f();
  printf(cadena);}

```

- Pantalla Inferior

**Ejercicio Interactivo 4 – Árboles Mini-Max*****Descripción:***

Las búsquedas con adversario en árboles binarios se han utilizado comúnmente para el desarrollo de videojuegos. Estas metodologías de recorrido de árboles son útiles para proporcionar soluciones de inteligencia artificial o bien calibrar el nivel de dificultad de aplicaciones de ocio en el ámbito computacional.

Este tipo de contenido se puede encontrar en las asignaturas de Metodología de la Programación (MTP) e Inteligencia Artificial e Ingeniería del Conocimiento, cuyos ejercicios se encargan de reforzar aspectos como éste. Los conocimientos que se quieren asentar son la resolución de problemas mediante el algoritmo Mini-Max y el método de poda Alpha-Beta. De esta forma, los ejercicios mostrarán determinados árboles binarios a resolver por los métodos citados, dando como pista el último nivel de dicho árbol resuelto. Pulsando sobre cada nodo del árbol, aparecerá en pantalla un teclado numérico con el que establecer el valor que el jugador considere oportuno. También se puede disponer del método de poda Alpha-Beta, aunque se ha considerado opcional en el desarrollo del modelo general.

El objetivo, como se puede deducir, es resolver el árbol Mini-Max. Esto implica indicar en cada nodo padre del árbol el valor que le corresponde, atendiendo a sus hijos. Si el nodo es MIN, se debe elegir el valor mínimo entre sus hijos. Si por el contrario el nodo es MAX, se deberá elegir el valor máximo.

Un nodo padre se desbloqueará cuando todos sus hijos tengan un valor asignado. El usuario puede habilitar, en caso de que no sea un examen, el modo “Supervisado” mediante el menú situado en la pantalla superior, el cual le ira avisando cuando comete algún error.

Percepción de Progreso y Realimentación:

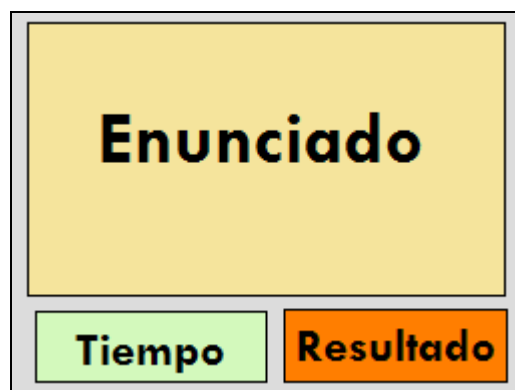
El jugador puede elegir la dificultad del ejercicio especificando la anchura del árbol y el grado de ramificación. Conforme obtenga resultados con éxito, estos parámetros se modificarán para aumentar la curva de aprendizaje.

Evaluación:

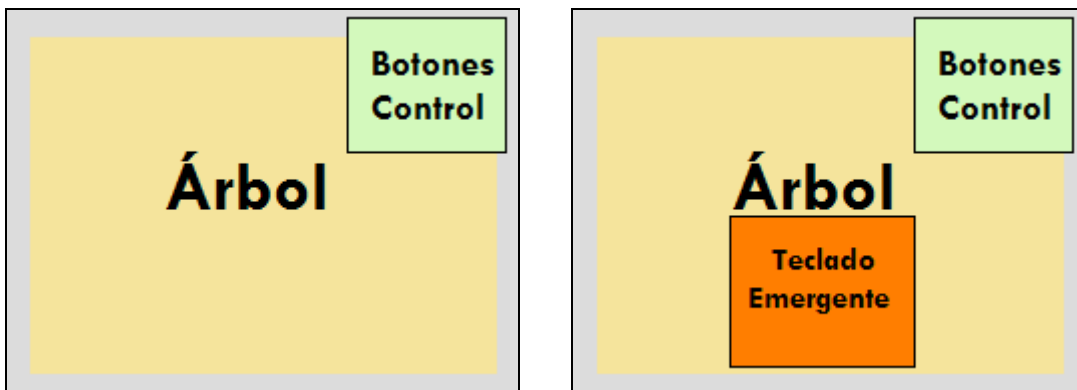
Un fallo común en los tipos de árboles Mini-max consiste en colocar un valor erróneo en un nodo. Para superar este ejercicio no se permiten más de 2 fallos, la complejidad de los árboles no es tal como para cometer más fallos. Además, este algoritmo es muy importante, como ya se ha comentado anteriormente para la creación de software de ocio.

Generalización del ejercicio:

- Pantalla Superior: contiene tres elementos de contenido, enunciado, tiempo y resultado.

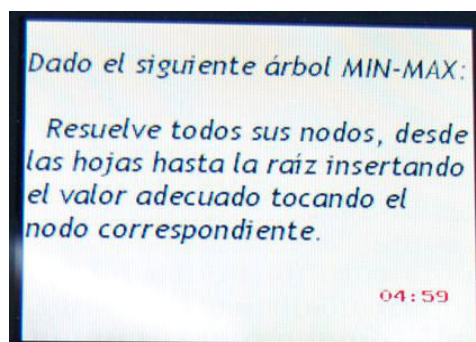


- Pantalla Inferior: La pantalla inferior está compuesta por tres elementos de interacción. Dos de ellos, siempre visibles, son el árbol y los botones de control, en cambio, el tercer elemento es un teclado donde introducir el valor del nodo en el que hemos pulsado, que aparecerá a petición del usuario en cada pulsación.

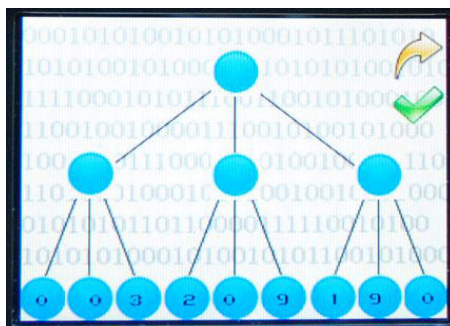


Instancia en la aplicación de este proyecto en la NDS:

- Pantalla Superior



- Pantalla Inferior



Ejercicio Interactivo 5 – Programación ROM / PAL

Descripción:

Hasta este momento, sólo se han tratado ejercicios del área de software. Con este nuevo tipo de ejercicios interactivos, basado en los dispositivos lógicos programables, se pretende reforzar el área hardware de la carrera de informática, con asignaturas como Fundamentos de Computadores. Los conocimientos que se adquieren y refuerzan son la resolución de módulos PLD (o también conocidos como *Dispositivos Lógicos Programables*), usados comúnmente para implementar sistemas combinatoriales.

El objetivo del ejercicio consiste en, una vez proporcionado el esquemático de puertas OR y AND y las funciones de conmutación a implementar, programar las conexiones según los datos que se proporcionan, tocando con el Stylus en la conexión correspondiente.

Percepción de Progreso y Realimentación:

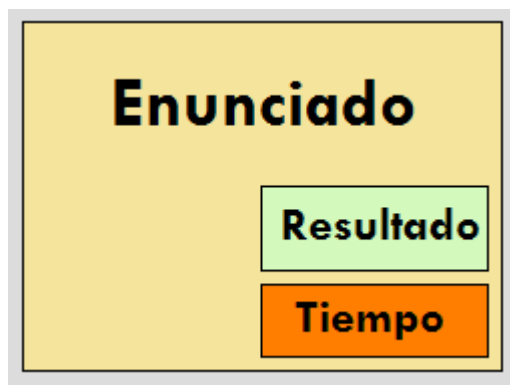
El alumno puede elegir el nivel de dificultad del ejercicio que va a realizar. Diferentes dificultades implican cambios en la complejidad de las funciones de conmutación (mayor número de variables y funciones).

Evaluación:

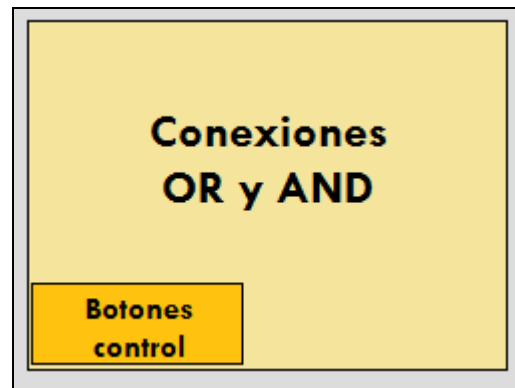
Cada conexión soldada (pinchada) por error se contempla como un fallo. Siendo bastante amplio el número de conexiones a soldar, los fallos permitidos para superar satisfactoriamente la prueba ascienden a 3 (remarcar que un módulo mal soldado no implementa las funciones que se piden en el ejercicio).

Generalización del ejercicio:

- Pantalla Superior: Consta de tres elementos de control. El enunciado cambia notablemente en cada nuevo ejercicio al generarse automáticamente nuevos casos.

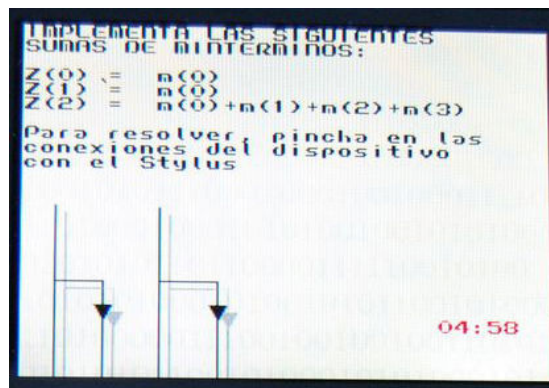


- Pantalla Inferior: Consta de dos elementos de contenido. El primero donde se encuentran las conexiones de las puertas OR y AND y el segundo los botones de control.

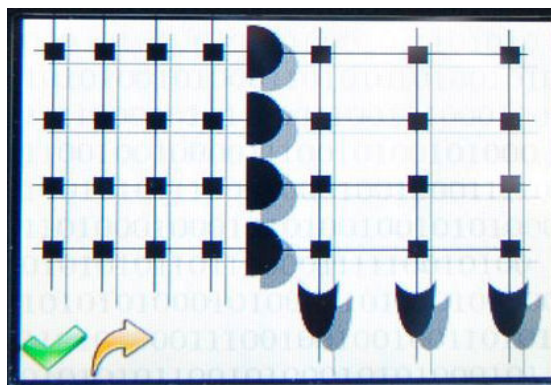


Instancia en la aplicación de este proyecto en la NDS:

- Pantalla Superior



- Pantalla Inferior



Ejercicio Interactivo 6 – Piezas lógicas

Descripción:

Siguiendo con el extenso área de hardware, otro modelo de ejercicios básicos en la carrera de informática son los referentes al conocimiento de circuitos eléctricos, ya que son la base de toda la parte física de la informática.

Las asignaturas que se pretenden reforzar con este tipo de ejercicio son las de Fundamentos de Computadores y Ampliación de Tecnología de Computadores, en cuyo temario se tratan las distintas puertas lógicas, su uso y funcionalidad.

Los conocimientos que se pretende que el alumno obtenga con la práctica de estos ejercicios son los referentes al uso correcto en determinadas situaciones de estas puertas lógicas. Se trata de que el alumno descubra qué puertas lógicas ha de usar en cada caso para que se produzca la salida correcta, desde las entradas hasta la salida especificadas.

El funcionamiento de estos ejercicios sigue el siguiente patrón: se proporciona un circuito incompleto en el cual, arrastrando los componentes lógicos que el usuario crea conveniente, se ha de conseguir que el circuito funcione de manera correcta, es decir, que a partir de las entradas del circuito se obtenga la salida correcta con los resultados de las puertas.

Percepción de Progreso y Realimentación:

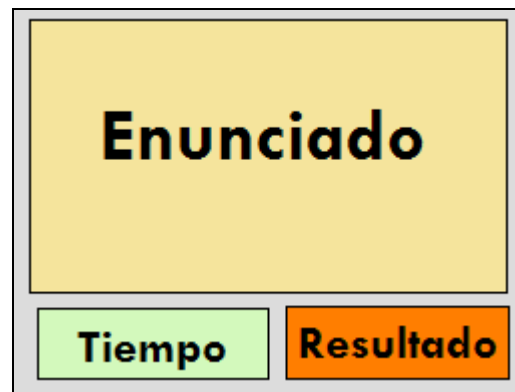
A medida que el alumno vaya avanzando en cada tipo de circuito, el mismo será un poco más complejo cuando vaya superando niveles, esto se consigue creando circuitos lógicos más extensos y complejos

Evaluación:

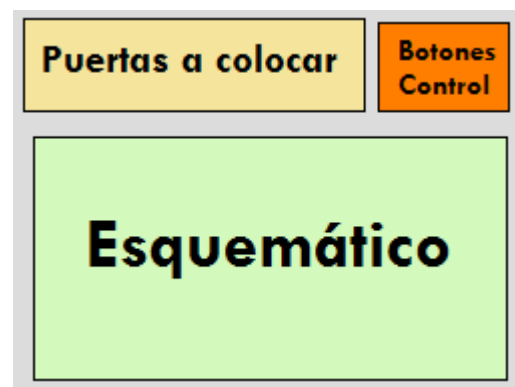
Un fallo se define en este ejercicio como una mala colocación de una puerta lógica, ya que no concordaría la salida con las entradas que tenga. El número máximo de fallos para superar este tipo de ejercicios es de dos.

Generalización del ejercicio:

- Pantalla Superior: Compuesto por tres elementos de contenido, el enunciado, que no suele cambiar en las instancias de este tipo de ejercicio, el tiempo que queda para terminar el ejercicio y la nota al corregir el ejercicio.

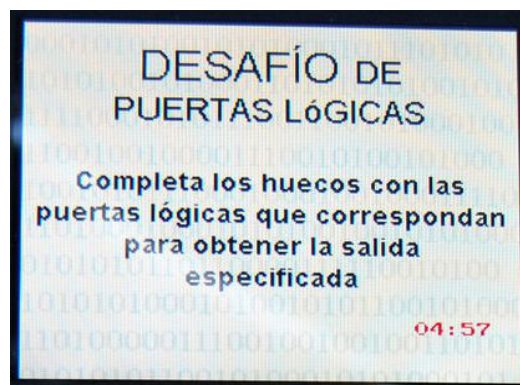


- Pantalla Inferior: En esta pantalla está compuesta por tres elementos de contenido, las puestas a colocar, los botones de control del ejercicio y el esquemático. Este último contiene los huecos donde colocar las puertas lógicas del primer elemento aquí descrito.

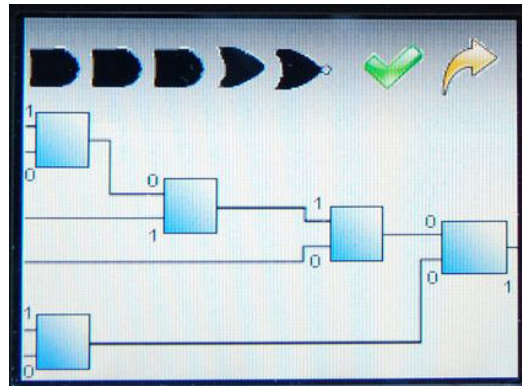


Instancia en la aplicación de este proyecto en la NDS:

- Pantalla Superior



- Pantalla Inferior



Ejercicio Interactivo 7 – Recorrido de Circuitos Sencillos

Descripción:

El recorrido de circuitos lógicos es algo propio de asignaturas como Fundamentos de Computadores, Laboratorio de Fundamentos de Computadores y Tecnología de Computadores. Con ello se adquieren, consolidan y refuerzan conocimientos que tratan sobre el funcionamiento de circuitos y puertas lógicas, en relación con el anterior tipo de ejercicio explicado.

El funcionamiento del ejercicio se describe como sigue: se muestra al alumno un circuito con los nodos de cada entrada y salida de las puertas lógicas que lo componen. El alumno debe pulsar en cada nodo para cambiar el valor de cero a uno, y viceversa, mientras considere oportuno. Su objetivo es que, dado un circuito compuesto por puertas lógicas y módulos combinacionales variados y a partir de una entrada conocida, se calcule el valor de cada salida que se genera.

Percepción de Progreso y Realimentación:

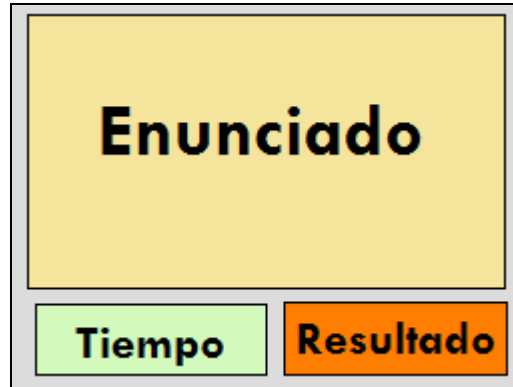
Se permite que el usuario elija la dificultad del ejercicio. Diferentes dificultades implican cambios en el número de puertas a atravesar, en el número de dígitos de la entrada, y en el aumento de complejidad que supondría incluir módulos combinacionales, o incluso secuenciales en un determinado ciclo de su ejecución.

Evaluación:

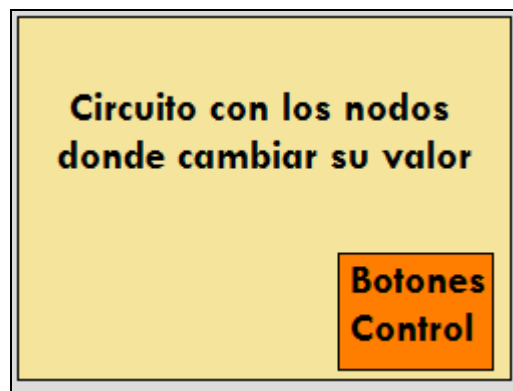
En este tipo de ejercicios un fallo se define como una errónea colocación del valor del nodo a la entrada o salida de alguna puerta lógica. El número máximo de fallos para dar el ejercicio como aprobado es de cinco, ya que los circuitos pueden llegar a ser considerablemente complejos.

Generalización del ejercicio:

- Pantalla Superior: Consta de tres elementos de contenido, el enunciado, que raramente se debe de modificar, el tiempo disponible para acabar el ejercicio y el resultado final al seleccionar la corrección del mismo.

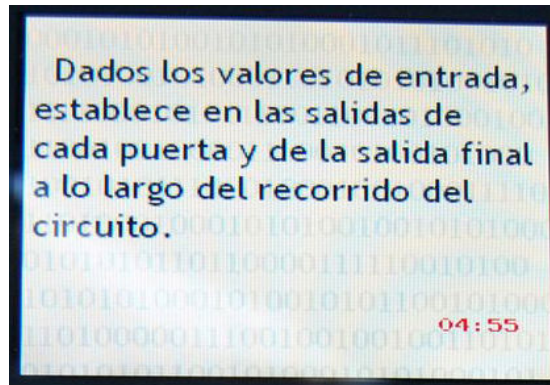


- Pantalla Inferior: Consta de dos elementos de interacción, el elemento que casi siempre aparece en todo ejercicio, que es el que contiene los Botones de Control y el elemento que contiene al circuito y los nodos donde se ha de pulsar para cambiar su valor.

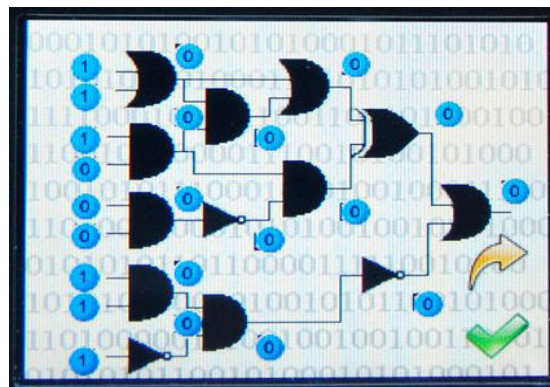


Instancia en la aplicación de este proyecto en la NDS:

- Pantalla Superior



- Pantalla Inferior

**Ejercicio Interactivo 8 – Mapas de Karnaugh*****Descripción:***

La asignatura que trata en profundidad los mapas de Karnaugh es la de Fundamentos de Computadores. Se trata de un método de simplificación muy útil para las funciones de conmutación de los sistemas tanto combinacionales como secuenciales, formado por un conjunto de cuadrículas en matriz, que representan los valores de cada una de las entradas.

El funcionamiento de este ejercicio se puede dividir en dos fases diferenciadas. La primera consiste en la resolución propia del mapa, es decir, se proporcionan una serie de minterminos a colocar en la cuadrícula adecuada en forma de 1s. Una vez colocados los que se consideren oportunos y pulsado el botón de corregir, se obtiene una puntuación provisional y se conduce a la segunda fase. Esta segunda fase se puede denominar como ‘fase de agrupación’, en la que figura

un nuevo tipo de botón ('Agrupar') con el que agrupar todos los conjuntos de mintérminos de cara a simplificar las funciones de conmutación.

El objetivo consiste en que dada una salida formada por un determinado de mintérminos, se simplifique su función de conmutación resultante a través de mapas de Karnaugh.

El jugador deberá rellenar el mapa con los 1s que obtenga de los mintérminos para cada valor de variable de entrada. Cuando crea haberlo completado, deberá agrupar esos 1s dibujando conjuntos con el lápiz táctil.

Percepción de Progreso y Realimentación:

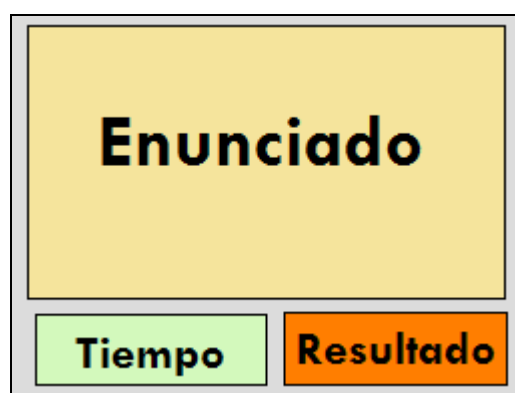
Se permite al jugador elegir la dificultad del ejercicio. Diferentes dificultades implican cambios en la complejidad de la tabla de verdad (más variables de entrada y salida equivale a mapas de Karnaugh más complejos).

Evaluación:

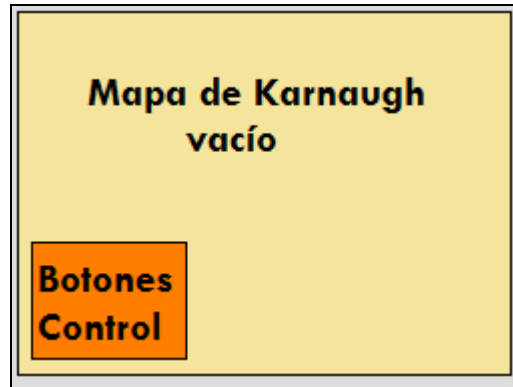
Un fallo en este tipo de ejercicios se considera un valor erróneo en alguna de las celdas del mapa de Karnaugh, ya que un valor equivocado cambia en gran medida las funciones de conmutación que puedan surgir del mapa. Para superar este tipo de ejercicios no se puede tener más de dos fallos. Las agrupaciones que se realicen no serán calificadas, puesto que en general, serán siempre soluciones válidas aunque su simplificación sea de menor calibre.

Generalización del ejercicio:

- Pantalla Superior: Aparecen tres elementos de contenido, el enunciado, con los mintérminos a colocar en el mapa de la pantalla inferior, el tiempo que falta para completar el ejercicio y los resultados del ejercicio al corregirse.

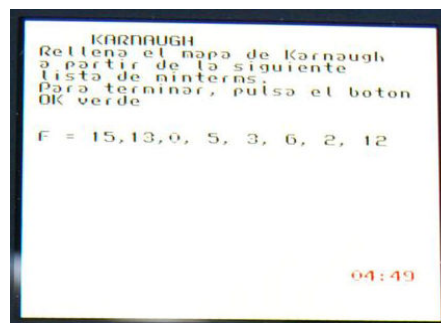


- Pantalla Inferior: Aparecen dos elementos de interacción. El primer elemento y el que más ocupa en la pantalla es el mapa de Karnaugh que en un principio aparecerá vacío. El segundo elemento es el que contiene los botones de control.

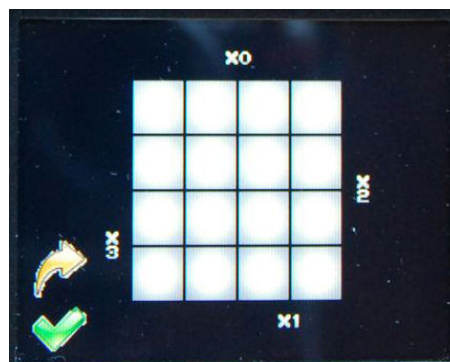


Instancia en la aplicación de este proyecto en la NDS:

- Pantalla Superior



- Pantalla Inferior



Ejercicio Interactivo 9 – Recorrido de intensidades

Descripción:

Otra faceta de estudio importante en el campo del hardware y diseño de circuitos eléctricos es el conocimiento de los parámetros físicos que los conforman, como los voltajes, las resistencias o, en este caso, la intensidad de corriente. La asignatura que cubre esta tipo de ejercicios es la Introducción a la Electrónica o la de Fundamentos Físicos de la Informática.

Los conocimientos que se pretenden reforzar con la práctica de este tipo de ejercicio son las características físicas de los circuitos eléctricos, más concretamente, los sentidos que la intensidad de corriente toma en diversos puntos del circuito.

El objetivo final consiste en que dado un circuito eléctrico, con datos de voltaje y resistencias, calcular el sentido de algunas de las corrientes eléctricas que atraviesan dicho circuito. Para ello el jugador deberá marcar las direcciones de cada corriente con flechas que determinen su sentido.

Percepción de Progreso y Realimentación:

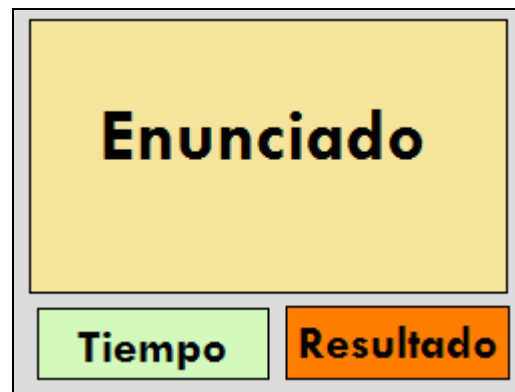
Se permiten al jugador elegir la dificultad del ejercicio. Diferentes dificultades implican cambios en los valores de las resistencias, los voltajes, en el número de ramas del circuito (con ello aumenta el número de intensidades a calcular) o en el añadido de componentes más complejos como transistores o diodos.

Evaluación:

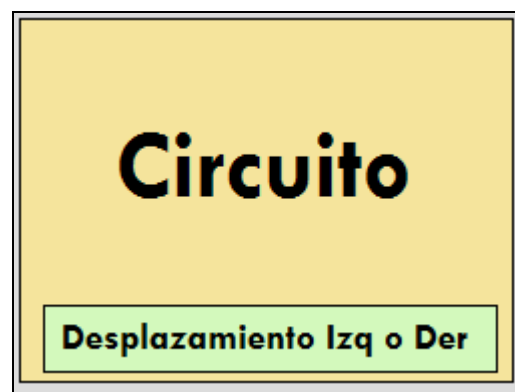
Se considera un fallo a la mala elección del sentido de la corriente. Para que un ejercicio de recorrido de intensidades sea considerado como bueno o aprobado no se deben de tener más de dos fallos.

Generalización del ejercicio:

- Pantalla Superior: Está compuesto por tres elementos de contenido, un enunciado, que no suele cambiar casi en ningún caso, el tiempo que le falta al ejercicio para que sea completado y el resultado que se produce al corregir el ejercicio.

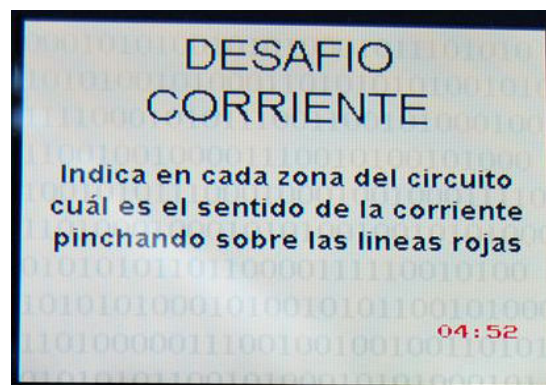


- Pantalla Inferior: Está compuesto por dos elementos de interacción. El que más destaca es la representación del circuito, que puede llegar a ser más grande que la pantalla de la propia Nintendo DS. De darse esta situación, con los botones de Desplazamiento Izquierdo o Derecho se facilitará la visión del circuito entero, haciendo un scroll horizontal.

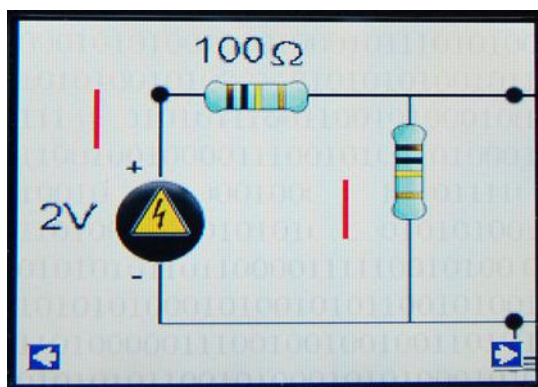


Instancia en la aplicación de este proyecto en la NDS:

- Pantalla Superior



- Pantalla Inferior



Ejercicio Interactivo 10 – Herencia y Polimorfismo

Descripción:

Volviendo a la rama de software, nos queda por cubrir un aspecto muy interesante de la programación orientada a objetos, el concepto de herencia. Este concepto permite la creación de objetos que aprovechen las funciones y características de una clase ya especificada, y añade las suyas propias. La asignatura que se pretende cubrir es la de Programación Orientada a Objetos (POO).

Los conocimientos que se desean reforzar y/o comprender con la práctica de estos ejercicios son los referentes a la herencia y el polimorfismo. El objetivo principal consiste en formar adecuadamente un árbol de herencia, como primer acercamiento a estos conceptos.

El jugador deberá, dado un conjunto de clases relacionadas, componer el árbol de herencia que las corresponda.

Los conceptos relacionados con el Polimorfismo no han sido desarrollados en esta primera aproximación.

Percepción de Progreso y Realimentación:

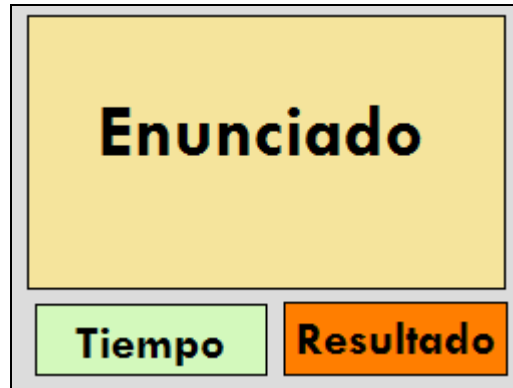
Una forma de subir la dificultad es mediante árboles más complejos o un mayor número de componentes de una clase.

Evaluación:

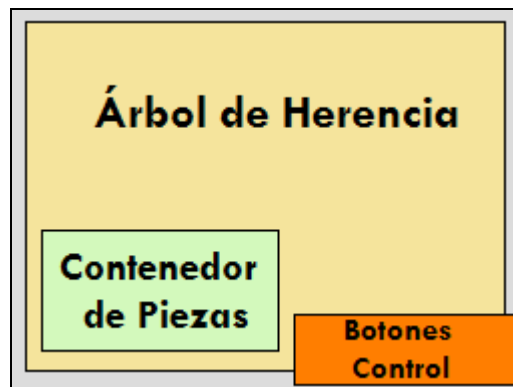
Un fallo es considerado como tal cuando una clase no esté colocada correctamente en el lugar que le corresponde. Para dar el ejercicio como bueno, se han de tener como máximo dos fallos, de lo contrario el ejercicio se considerará suspenso.

Generalización del ejercicio:

- Pantalla Superior: Consta de tres elementos de contenido, un enunciado, en el que se especifica la herencia a formar con las piezas de la pantalla inferior, el tiempo y el resultado, que se mostrará cuando se corrija el ejercicio.

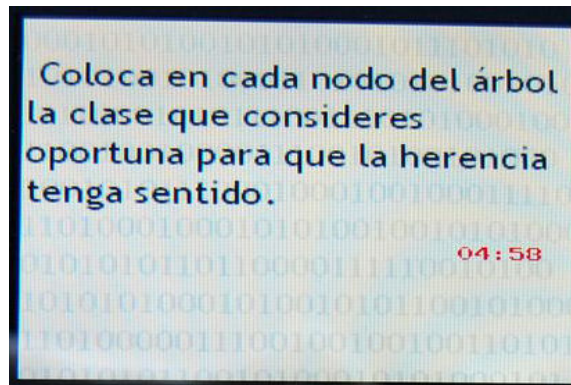


- Pantalla Inferior: Consta de tres elementos de interacción. El 'árbol de herencia' que contiene los huecos donde se deberán situar las piezas del árbol, las cuales se encuentran en el contenedor de piezas y, por último, el elemento que contiene los botones de control..

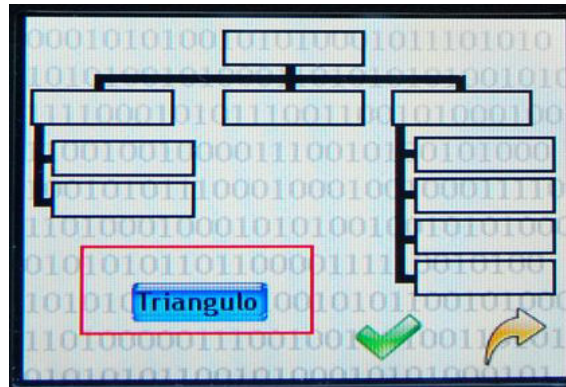


Instancia en la aplicación de este proyecto en la NDS:

- Pantalla Superior



- Pantalla Inferior



6. Arquitectura General

Para responder adecuadamente al reto tecnológico que supone desarrollar un binomio de videojuego educativo y LMS en la plataforma Nintendo DS, y siguiendo el modelo conceptual descrito en el apartado 4 de este mismo documento, la arquitectura general de CS-Training se ideó desde un principio teniendo en cuenta los siguientes aspectos relevantes:

- La arquitectura de la aplicación debía ser totalmente escalable: Para ello, cada instancia de ejercicio interactivo se encuentra encapsulada dentro de un cargador global. De esta forma, añadir nuevas instancias a partir de los modelos actuales es un proceso sencillo y directo.
- El proceso de evaluación debía llevarse a cabo en la aplicación: Cada modelo de ejercicio interactivo dispone de un evaluador local a través del cual obtiene su puntuación. Todas las puntuaciones obtenidas se recogen en una ponderación que representa la puntuación final, de lo que se encarga un evaluador global independiente.
- La comunicación debía establecerse en dos vías: La aplicación envía a través de la conexión inalámbrica los datos inherentes a cada perfil de ejecución. El LMS configura parámetros relativos al desarrollo de las ejecuciones.
- Moodle debía seguir siendo funcional a todos los niveles: La comunicación se establece sin modificar las características y funcionalidades propias de Moodle como sistema abierto a todo tipo de plataformas (p.ej. ordenadores personales).

Todas estas características son las que ayudan a los recursos de que dispone CS-Training a asemejarse al modelo de objetos educativos.



Figura 6.1 Vertientes de diseño en CS Training

Debemos distinguir dos vertientes de diseño bien distintas: Por un lado, la aplicación independiente ejecutable en Nintendo DS, y por otro, el conjunto formado por la aplicación en comunicación con nuestro LMS, Moodle.

Visión de Alto Nivel (Aplicación)

A grandes rasgos, CS-Training como videojuego educativo se estructura atendiendo a sus modos de juego y los contenidos de cada uno. Tanto la práctica libre como la evaluada comparten su acceso a las instancias de los minijuegos, que se organizan en dos conjuntos conceptuales distintos: la rama de software, y la de hardware, siguiendo la línea de las asignaturas de la Ingeniería Superior en Informática.



Figura 6.2 Una visión de alto nivel de la organización de CS Training

El esquema muestra las diez instancias de ejercicios que acompañan a CS-Training desde un principio, cinco de cada conjunto.

En una visión más detallada, a continuación se describen el diagrama de clases y los esquemas internos de la arquitectura general de CS-Training.

Diagrama de clases (UML)

Cargador

La clase principal y esqueleto de la aplicación CS-Training es el Cargador. Es la encargada de ejecutar cada instancia de juego, dentro de los modos de uso disponibles, así como de generar y comunicar las evaluaciones a través de la conexión WiFi. Toda la estructura de la aplicación se sostiene sobre esta clase principal, cuya multiplicidad es única para las relaciones que posee con las otras clases. Las funciones que contiene se detallan a continuación:

Menus(): Como su nombre indica, se encarga de generar los distintos menús que aparecen a lo largo de la aplicación, desde la introducción hasta el entrenamiento, y su navegación.

Evaluacion(): Realiza todo lo relacionado con el modo de tests evaluados, comenzando por una autenticación en el servidor del LMS, y terminando con el guardado de la nota final y un resumen de la ejecución en la base de datos del LMS.

AnalizadorDeCadenas(): Se precisa de un analizador de las cadenas de configuración de los tests para, en el momento de leer dicha información, generar un vector de instancias de los ejercicios interactivos que más tarde se realizarán y evaluarán. De esto se encargará esta función.

Login(): Realiza la autenticación del usuario en Moodle, para después mostrar los cursos y tests a los que puede tener acceso. Un alumno estará matriculado en un determinado curso y dispondrá de los tests que el profesor haya habilitado al efecto.

Temporizacion(): Esta función define la interfaz básica del temporizador, con medidas en minutos y segundos, que usará cada instancia de juego.

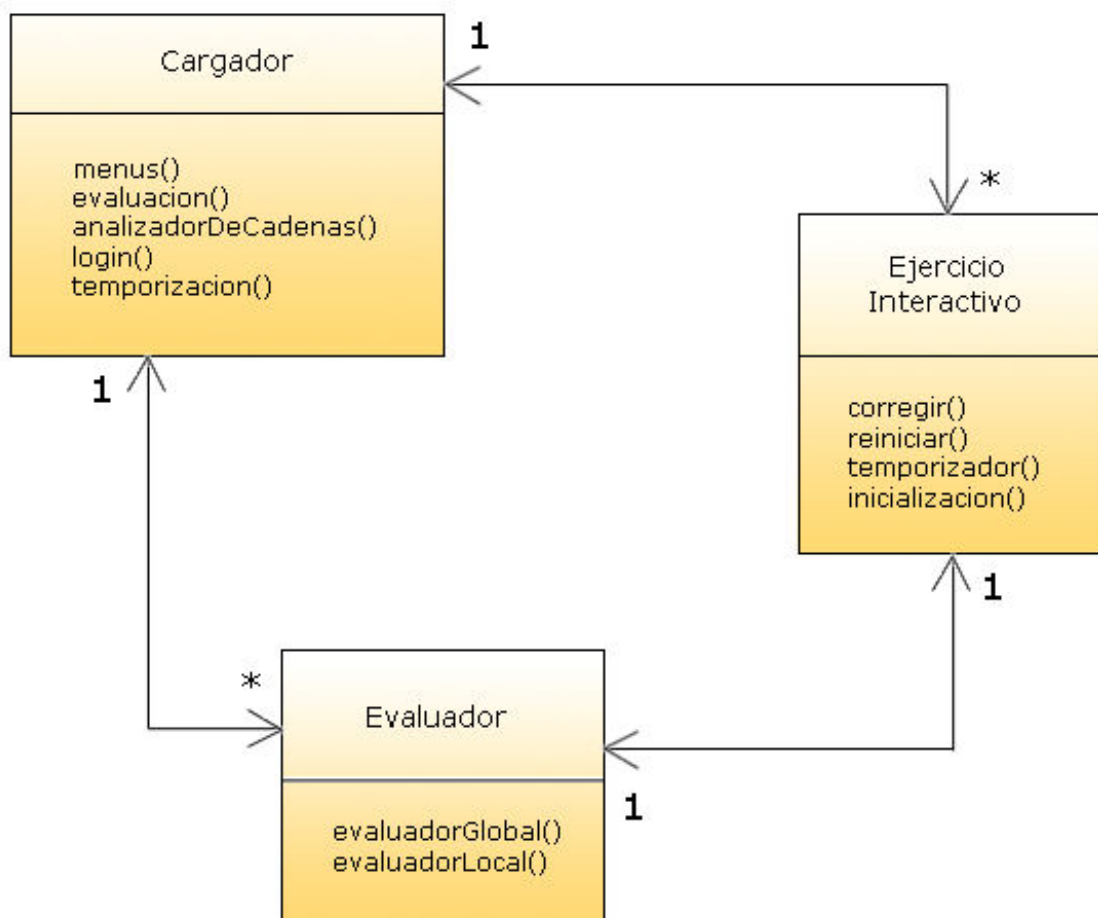


Figura 6.3 Diagrama UML que representa la arquitectura software de CS Training

Ejercicio Interactivo

Otra clase con gran protagonismo en la arquitectura de CS-Training es la que da forma al modelo de ejercicio interactivo del que beberán las instancias que nos encontremos a la hora de jugar. Cada ejercicio interactivo dispondrá de unas características de interacción y desarrollo específicas, aunque su estructura siga el mismo patrón de funciones y control. Las siguientes funciones generalizan el funcionamiento de cada ejercicio:

Corregir(): Haciendo uso del evaluador local del que disponga el modelo de ejercicio interactivo, se obtendrá una calificación siguiendo la lógica descrita en el apartado anterior del documento. Se requiere de una relación unitaria entre las clases evaluador y ejercicio, ya que para cada ejercicio existe un (uno y sólo uno) tipo de evaluador local, y viceversa.

Reiniciar(): Sólo en el caso del modo de entrenamiento, el usuario dispondrá de esta función que inicializa de nuevo todos los parámetros de la instancia actual, incluida la calificación y el temporizador.

Temporizador(): A partir de la interfaz definida en la clase Cargador, se genera un cronómetro que pondrá un límite al tiempo de resolución de cada ejercicio. El tiempo total es de 300 segundos, es decir, 5 minutos.

Inicializacion(): Todo modelo de ejercicio interactivo tendrá una primera fase con la ejecución de esta función, en la que se crearán sus propios elementos de interacción y control.

Evaluador

Por último, tanto el cargador como los ejercicios precisan de las herramientas necesarias para obtener las calificaciones locales y global, definidas en la clase que nos ocupa. Sólo existirá un evaluador global, con la labor de reunir todas las calificaciones en una ponderación, media aritmética en este caso, de acuerdo al método habitual de evaluación en la enseñanza académica. Por otro lado, al estar encapsulados los ejercicios interactivos bajo un evaluador local cada uno, existirán tantos como modelos de ejercicio haya, con las características de evaluación necesarias para el modelo en cuestión.

NOTA: Las herramientas de desarrollo de software para Nintendo DS utilizadas en la creación de CS-Training están basadas en el lenguaje de programación C, carente de orientación a objetos. En este sentido, el diagrama de clases explicado con anterioridad es una aproximación al modelo real de la aplicación, por motivos de claridad y estandarización.

Esquemas Concretos

Ya hemos explicado de manera superficial cómo se organiza la aplicación, incluyendo un diagrama de clases con la estructura del código. Seguidamente, abordaremos los aspectos referentes al diseño concreto de la aplicación independiente, y del conjunto en comunicación con Moodle, las dos características más innovadoras de CS-Training.

Aplicación de Nintendo DS

La arquitectura de CS-Training se constituye sobre un cargador general (loader), formado por las distintas instancias de ejercicios interactivos que se incluyen encapsuladas junto a su evaluador local, contenidas todas a su vez en un evaluador global. Este último, a petición del cargador, será el encargado de tomar los resultados que produzcan los evaluadores locales y generar la calificación final, que será enviada a Moodle como se explica en el esquema del apartado siguiente.

Con esta estructura independiente se pretende dar soporte a futuras ampliaciones, en las que se incluirían nuevas instancias y/o modelos junto a sus evaluadores en una aproximación sencilla y directa, incluso automatizable. Todo ello refuerza aún más la idea de que estamos tratando con objetos educativos reutilizables y ampliables, cuya compatibilidad no se limita sólo a este proyecto.



Figura 6.4 Esquema de la arquitectura de la aplicación

Comunicación e Integración con un LMS

La componente más innovadora de CS-Training se corresponde con la posibilidad de añadir un mayor soporte a la arquitectura estándar de la aplicación mediante un Sistema de Gestión del Aprendizaje, en nuestro caso, Moodle.

Moodle (en su interfaz) hace uso de un servidor web, y además contiene una base de datos en la que se almacenan todos los datos referentes a cursos, alumnado, profesorado, etc. y a la que se accede y modifica mediante unas funciones en lenguaje php habilitadas en el servidor para ello. De esta forma, haciendo uso de unas nuevas funciones sobre la base de las ya existentes (de cara a dar soporte a la plataforma Nintendo DS sin que se vea afectada la integridad de Moodle), establecemos las características del tipo de comunicación que se llevará a cabo entre ambas plataformas a través del servidor web. Distinguiremos dos vías en esta comunicación, con un orden de ejecución bien definido:

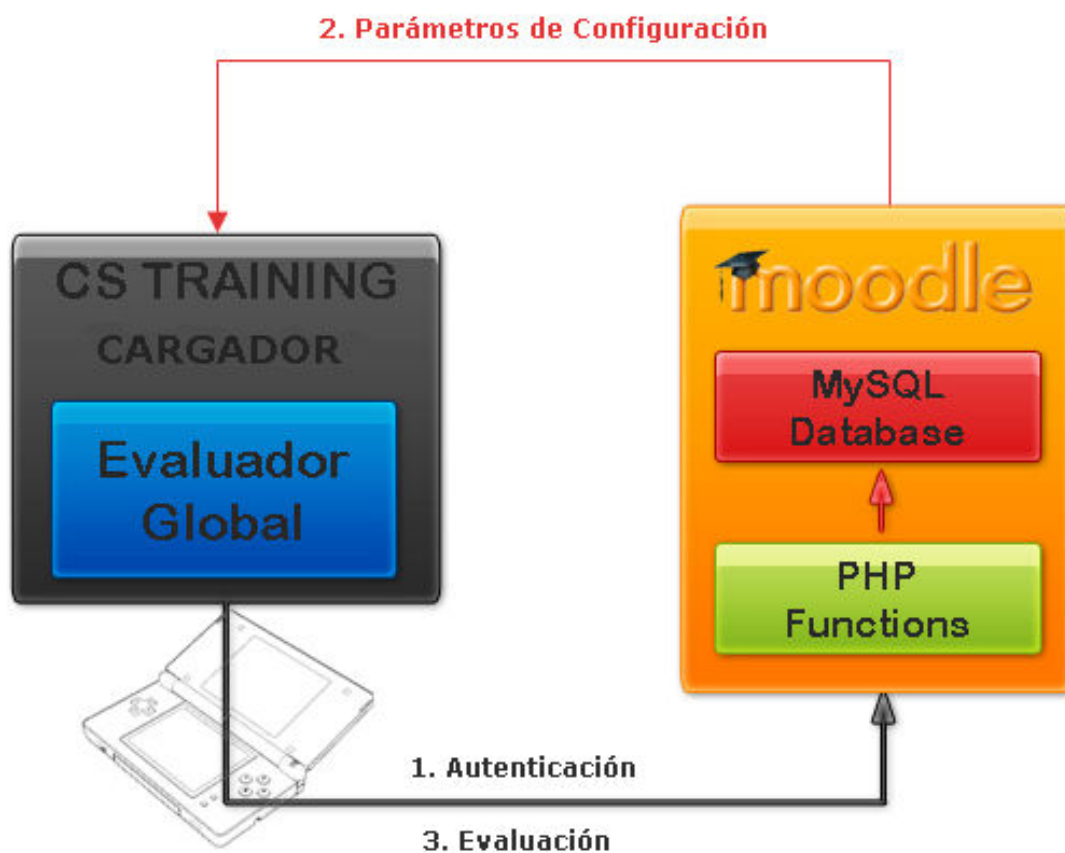


Figura 6.5 Esquema de comunicación entre NDS y Moodle

1. **Autenticación** (de Nintendo DS a Moodle): En primer lugar, el usuario ha de autenticarse mediante un proceso sencillo en la aplicación, de forma que se conozcan al instante su perfil y sus cursos matriculados, y se asigne su rol.
2. **Parámetros de Configuración** (de Moodle a Nintendo DS): En segundo lugar, se establece una vía de comunicación en sentido Moodle-NDS, en la que se permite configurar los tipos de instancias que se incluirán en el test evaluado a

realizar (incluyendo aspectos como la dificultad del ejercicio) a partir del test que haya seleccionado el usuario. Gracias al analizador de cadenas del módulo Cargador, la aplicación determina todos esos datos transmitidos desde el LMS.

3. **Evaluación Final** (de Nintendo DS a Moodle): Toda vez que se obtenga la calificación ponderada de la realización del test, ésta se transmitirá a Moodle haciendo una nueva inserción en la base de datos, más concretamente, en la tabla habilitada para almacenar los intentos que el alumno realiza. Se envía además información referente al tiempo consumido en la realización, y a la nota individual de cada instancia de ejercicio.

Las transmisiones de información se realizan siempre en un orden estricto, puesto que, por ejemplo, el resultado de la ejecución no se conoce hasta el final de la comunicación, y un usuario ha de autenticarse para poder saber con certeza los tests a los que tiene acceso.

7. Proceso de desarrollo

Análisis inicial

En un primer análisis del campo de aplicación del desarrollo, surgieron diversas ideas con el fin de abarcar el mayor área de estudio posible. Tratándose de un videojuego educativo, era pertinente elegir una categoría didáctica concreta, y determinar cuáles serían los conceptos a practicar/aprender y cómo se evaluarían. Las primeras ideas en surgir iban desde una autoescuela virtual hasta los tests psicológicos que se realizan en los institutos. Pronto se inclinó la balanza hacia los conceptos reales que se imparten en las aulas de las universidades de nuestro país, puesto que representaban idealmente lo que se iba a practicar o reforzar con nuestra aplicación y su método de evaluación era directo. Y qué mejor campo de estudio, que el que hemos tenido al alcance estos últimos años: nuestra propia carrera universitaria. La idea ya había sido escogida, y por unanimidad.

Estudio del dominio

Dado que estudiamos Ingeniería Informática, contamos con el acceso a los conceptos y recursos fundamentales para el estudio de las asignaturas que hemos cursado. Esto nos ha ayudado a la hora de crear los ejercicios y desafíos (basados escrupulosamente en los que se practican en clase) dirigidos a otras personas, ya sean de nuestra carrera, o las que en un futuro desean iniciarse en el mundo de la informática.

Análisis de riesgos

- Librerías PAlib de código abierto ("Open Source"): No están oficialmente soportadas, por lo que su funcionamiento no está garantizado para todos los casos posibles. Aunque sacan versiones nuevas de las librerías utilizadas en este proyecto, desde el principio se decidió utilizar la versión vigente hasta ese momento.
- La memoria de Nintendo DS es muy limitada: Las librerías no gestionan todavía la memoria de la consola de una manera tan eficiente como lo realiza el SDK oficial de Nintendo. Es por esto que, al usar fondos de pantalla, sprites o sonidos, se debe tener cuidado en no saturar la memoria, hasta cierto punto que lleve a un mal funcionamiento de la aplicación, o incluso a su carga defectuosa.
- No se dispone de un acceso total a los dos microprocesadores que posee la consola: La Nintendo DS consta de 2 procesadores, un ARM7 y un ARM9, cada uno con sus funcionalidades características, ya sea el WiFi o la gestión eficiente de la memoria. En las

librerías utilizadas, solo se dispone de acceso al procesador ARM7, con menor memoria y encargado de funciones más sencillas.

- La aceleración 3D que se pensó usar en un principio no era factible: el OpenGL implementado para estas librerías es muy limitado, faltaban funciones primitivas para crear un buen entorno de aplicación.
- La comunicación vía WiFi está en un estado prematuro: Las funciones encargadas de dar soporte a la conexión inalámbrica de Nintendo DS en las librerías utilizadas no contaban con la funcionalidad deseada, por lo tanto la configuración del punto de acceso queda restringida a una aplicación externa y la transmisión de información se realiza a través del protocolo HTTP.

Organización del modelo de proceso

Las ideas que se tuvieron a la hora de estructurar este proyecto, atendiendo a los roles de las plataformas que intervienen en CS-Training, fueron las siguientes:

- ***Nintendo DS se encarga de todo***

Con una sola ROM, todos los ejercicios interactivos están en la consola Nintendo DS. Por lo que se encargará de hacer todos los cálculos de las notas, tanto globales como locales a cada ejercicio interactivo. Finalmente, después de todos los cálculos y desarrollo de los ejercicios por parte del alumno, se envía la nota final ya calculada para que sea almacenada en Moodle.

Ventajas:

- ✓ El tráfico de información es mínimo
- ✓ Aunque no existe conexión a internet, el usuario puede darle uso a la aplicación

Inconvenientes:

- ✓ Todo está en la NDS, para cualquier cambio o actualización es necesario tocar el código
- ✓ Muy difícil hacer expansiones de la aplicación

- ***Moodle se encarga de todo***

Moodle posee en su base de datos todos los ejercicios interactivos, los cuales son descargados a la Nintendo DS a demanda del usuario. La NDS únicamente toma un papel de interfaz con el usuario y usa el cargador para ejecutar los ejercicios interactivos, por lo tanto, todas las notas son calculadas en Moodle.

Ventajas:

- ✓ El servidor tiene más capacidad de proceso que la NDS
- ✓ La ampliación y actualización de la aplicación se hace más sencilla

Inconvenientes:

- ✓ Puede provocar un gran tráfico de información a la hora de bajar los E.I.
- ✓ El cargador de la NDS es muy complejo
- ✓ Posible colapso del servidor a la hora de una gran demanda de información

- ***Modelo híbrido entre Nintendo DS y Moodle***

Se reparten las tareas ambos sistemas de manera equitativa. Parte del cálculo se realiza en Moodle y parte en la consola, en la cual están los módulos de los juegos, los cuales son configurables mediante cadenas de texto enviadas desde Moodle.

Ventajas

- ✓ Compensa el grueso de la transferencia de información entre el servidor y la NDS
- ✓ Se comparte el cálculo entre ambos sistemas

Inconvenientes:

- ✓ Sigue siendo poco flexible a la hora de hacer alguna actualización de la aplicación de la NDS

Métricas del proyecto

Tamaño estimado del código: 20.000 líneas de código

Número de Clases: 45 clases

Número de PHP: 10 ficheros para parchear el LMS

Número de gráficos generados en la Nintendo NDS:

- Fondos de pantalla: 44 imágenes
- Sprites: 198 imágenes

Desarrollo en iteraciones

Se ha llevado a cabo un modelo de micro-iteraciones durante todo el año que ha durado el proyecto, realizando hasta tres iteraciones principales, con entregas semanales para cada una. En la primera, de recopilación de ideas e información, además de hacer la pruebas pertinentes con el hardware y el software que disponíamos y saber hasta dónde podíamos llegar. Por lo tanto, se marcó como objetivo inicial llegar a la especificación de 10 juegos o ejercicios interactivos y el diseño de los mismos. También se instaló Moodle para su uso y conocimiento, que junto dos implementaciones de ejercicios interactivos que guardaban resultados de exámenes en Moodle hicieron despuntar el proyecto. Finalmente, se empezó a estudiar la posibilidad de dar soporte para SCORM, pero se desechó al inicio de la segunda iteración por incompatibilidades.

En la segunda, la mayor parte del trabajo se centró en el cargador de juegos de la NDS, tratando los perfiles, los niveles de dificultad y la interfaz con el usuario. Se buscó de la misma manera, la implementación de, al menos, 5 ejercicios interactivos. Toda la información que se iba generando del proyecto se iba documentado para poder ser utilizada en este documento.

Finalmente, en la tercera iteración, se busca la implementación del resto de ejercicios interactivos y la mejora y solución de problemas. También la terminación de la memoria del proyecto, un documento elaborado, con un gran número de iteraciones.

A continuación, con todo detalle, se describe el transcurso del proyecto, dividiendo cada iteración en las semanas que se tuvo reuniones y se hubieron marcado objetivos.

Primera Iteración

- Semana I

Una vez que el grupo tuvo claro el objetivo que iba a seguir, se buscaron 15 propuestas de mini-juegos que tuvieran que ver con los diferentes aspectos de las carreras de informática que se imparten en la Universidad Complutense de Madrid. Finalmente, se seleccionaron 10 mini-juegos para realizar su generalización y posterior instancia durante todo el curso.

Por otro lado, también se estuvo investigando sobre el funcionamiento del LMS a utilizar, en este caso, Moodle, por lo que se estudiaron diferentes manuales, pero sobre todo, basándose en la página oficial de dicho LMS.

- Semana II

A la hora de realizar las descripciones de los mini-juegos, cada componente del grupo utilizó un formato diferente, por lo que se propuso y llevó a cabo una estandarización de las descripciones de los ejercicios para que su futuro desarrollo fuera más inmediato.

Siguiendo el trabajo de estudio con Moodle, se llevó a cabo su instalación y las pruebas que se citan a continuación.

- Creaciones de cursos de prueba, profundizando en las tablas MySQL, cómo se generan y cuál era su estructura
- Creación de usuarios, tanto administradores, como profesores o alumnos de prueba, estudiando la generación y estructuras de las tablas de la base de datos de los mismos.
- Por último, se investigó cómo introducir una nota de algún curso creado de un alumno de prueba, no de manera común, que sería haciendo algún ejercicio, si no accediendo directamente a la Base de Datos.

- Semana III

Con todo el software base que se necesitaba para trabajar, se avanzó en los mini-juegos. Se decidió empezar a profundizar en el tipo de juego 'Programación ROM / PAL', por lo que su documento de descripción se elaboró más y se subió a la wiki, herramienta que se empezó a utilizar para informar a los componentes del grupo y a los profesores directores sobre los avances que se producían.

Por otro lado, se inició el desarrollo del prototipo del tipo de juego 'Programación ROM / PAL', consiguiendo grandes y espectaculares resultados que motivaron al grupo para continuar con el proyecto.

- Semana IV

El prototipo había sido un éxito, pero se necesitaba refinar mucho para que fuera realmente funcional y se tuviera una interacción apropiada, por lo tanto se terminaron las partes de diseño y funcionalidad con el Stylus.

En el ámbito del LMS se buscaba la posibilidad de poner una nota de manera automática sin acceder a la propia aplicación Moodle.

- Semana V

Se terminó el tipo de juego 'Programación ROM / PAL'; esto quiere decir que se podían practicar varios niveles de dificultad y enviaba notas al LMS, en el modo de evaluación, el cual las almacenaba en su Base de Datos.

En ésta misma micro-iteración fue posible la creación de la instancia del tipo de juego 'Puzzle Programación', la cual se terminó con la misma funcionalidad que el tipo de juego citado antes, es decir, pudiendo practicar varios niveles de dificultad y enviando notas de este tipo de ejercicio al LMS en el modo de evaluación.

Para que se pudieran hacer la demos de una manera mucho más automática, se procedió a instalar el LMS en un portátil.

- Semana VI

En esta semana se produjo una de las primeras entregas de todo el trabajo de la aplicación realizado hasta ese momento, es decir, el desarrollo en la Nintendo DS y en Moodle y cómo interactuaban entre ellos. La entrega fue un éxito, los juegos en la Nintendo funcionaron como deberían y la comunicación con el LMS fue perfecta.

- Semana VII

En esta última semana se estuvo trabajando en un primer boceto de este documento, la memoria del proyecto. Aún le faltaba mucho contenido, pero con los consejos de los profesores directores se pudieron arreglar varios apartados.

Segunda Iteración

- Semana I

Después de los consejos dados por los profesores directores sobre la memoria, la corrección de la misma consistió en una reorganización de todos los apartados y de los contenidos que había, que se tuvieron que completar o rehacer casi todos.

Una vez resuelto el asunto de la memoria por el momento, se continuó con la planificación general del proyecto. Se hizo especial hincapié en los objetivos de la segunda iteración, como un estudio de campo sobre un soporte SCORM para nuestra aplicación, el cual fue desechado por incompatibilidades en el sistema.

También se fue conceptualizando un cargador en la Nintendo DS para la ejecución de todos los tipos de mini-juegos que se fueran a desarrollar durante el curso, se encontraron las posibilidades de que una ROM fuera capaz de ejecutar otra ROM o bien todo fuera integrado en la propia aplicación de la Nintendo DS. Por último, se fijó el objetivo de la implementación de cinco tipos de juegos al acabar esta iteración, que se elegirían de forma aleatoria

- Semana II

En esta semana se tomó la decisión de cómo iba a ser finalmente el cargador de la aplicación de la Nintendo DS, que consistiría en dos modos de funcionamiento:

- Normal: Serviría de entrenamiento para el usuario y no influiría en la nota, es decir, sus resultados no serían relevantes.
- Test: Compuesto por una batería de ejercicios, se correspondería con los exámenes que pudiera realizar el usuario.

Dentro de los propios mini-juegos, se diferencié el tipo de evaluador, se generalizó y se decidió que aceptara muchos parámetros de entrada para configurarlo. Los posibles tipos de evaluador que se sacaron en claro fueron por pulsación y por piezas.

Sobre los mini-juegos, se insertó un nuevo concepto para denominar a los ejercicios propuestos al alumno, el de 'Ejercicios Interactivos', que tendría un cierto tiempo de realización que sería mostrado en alguna de las pantallas de la consola Nintendo DS.

Respecto al ámbito de Moodle, se sostuvo en un principio que el sistema eligiera los juegos que evaluaban al alumno, pero se desechó esa idea. La mejor solución era tener un perfil y un nivel, por lo tanto con estos parámetros el cargador de la NDS sabría qué acción efectuar. Dentro de este perfil se asignan los campos que correspondan, en este caso, podrían ser Campo Software y Campo Hardware. Para cada test se le asignaría un conjunto de notas, que serían el resultado de la batería de ejercicios.

Con todo lo visto, se realizó una planificación de tareas específicas durante la segunda iteración:

- Un componente del grupo se encargaría del cargador de la consola Nintendo DS.
- Otro componente del grupo se encargaría del tipo de juego de los árboles y evaluadores.
- Finalmente, se repartiría el trabajo para que un juego estuviera totalmente reforzado de cara al final de la iteración.

Se siguió trabajando en la Memoria, se empezaron a usar formatos y esquemas.

- Semana III

En esta semana hubo una pequeña entrega para valorar los avances del proyecto en la Nintendo DS, y se concluyó que era un buen prototipo funcional, aunque existían ciertos matices de los que ocuparse. Fue a nivel de estructura de la aplicación donde había que realizar un esfuerzo especial para encontrar una arquitectura eficiente para el proyecto. Se debía crear un módulo para el evaluador que aún no aparecía por ningún lado. Esto conllevaba a tener que hacer el evaluador general y los locales encapsulados bajo cada ejercicio interactivo.

En el caso del almacenamiento de algunos datos en la NDS, se tenía por seguro que habría que guardar el usuario, la clave y los cursos de los que se estuviera evaluando. A la hora de evaluar se requerirían los tres datos: usuario, clave y curso. Había que intentar que se generara un gráfico con los progresos que iba teniendo el alumno.

Después de cualquier avance habría que subir los datos a las herramientas de SVN y a la Wiki.

Finalmente, en esta micro-iteración, se marcaron ciertos objetivos para la siguiente:

- Profundizar en el ámbito de Moodle y el funcionamiento de las tareas
- Posibilidad de configurar la WiFi de Nintendo DS desde nuestra aplicación
- Tener en el SVN siempre la última versión
- Subir la documentación a la Wiki con la planificación de cada semana

- Resolver el método de autenticación desde la Nintendo DS y definir el evaluador global para los ejercicios interactivos.

- Semana IV

Con los buenos resultados obtenidos en la micro-iteración anterior se repitieron algunos objetivos antiguos con nuevos alcances, ampliación de los ejercicios interactivos desarrollados hasta el momento y profundización en Moodle sobre la manera de tratar su base de datos.

Apareció un nuevo planteamiento para otro tipo de ejercicio, juegos con objetivos de grupo. Se buscaba que un conjunto de alumnos, con un objetivo común dividido en varias partes, trabajaran en equipo con un líder o bien con una jerarquía horizontal. Finalmente, se desechó la idea por la actual limitación de las librerías y el tiempo para un desarrollo y una investigación adecuados.

- Semana V

Siguiendo con la modularización de la aplicación de la Nintendo DS, los evaluadores locales, para juegos actuales, se hicieron independientes al resto para facilitar su modificación o reemplazo en el caso de un cambio drástico de corrección.

En esta misma semana se probó también el desarrollo del cargador de los ejercicios interactivos, pero se sacaron más errores o malas funcionalidades de lo que se esperaba, por lo que habría que hacer un esfuerzo especial en su depuración.

Dentro del ámbito del LMS, acerca de cómo funcionarían los cuestionarios o exámenes, la duda quedó resuelta, es decir, habría que comunicar varias notas una vez corregidos los ejercicios interactivos realizados por el alumno.

- Semana VI

En esta semana se tuvieron que perfeccionar los evaluadores locales, y el global si los anteriores ya estaban creados. Se continuó con una constante depuración del cargador, ya fuera por motivos estéticos como elementos de interacción y control o motivos funcionales, como la corrección de los botones de rehacer y corregir en la aplicación de la Nintendo DS. Finalmente, en el ámbito de Nintendo DS, se llevó a cabo la implementación de la comunicación de varias notas, una por cada intento.

Debido a la gran cantidad de información que se estaba subiendo a la Wiki del proyecto se tuvo que hacer una reorganización general de los contenidos.

Por último, en la comunicación entre la consola y el LMS se tuvieron que buscar vías para resolver la autenticación desde la Nintendo DS.

- Semana VII

Se consiguió implementar el login y el guardado de sus datos, obteniendo ID de usuario, pero se intentaron buscar vías para mantener la sesión sin cookies (datos de configuración que suelen almacenarse en nuestro navegador, pero que era imposible en la Nintendo DS), las cuales usaba el LMS para mantener la sesión.

Dentro de la aplicación de la Nintendo DS se terminó el tipo de juego de 'Árboles Minimax', con la modularización que se especificó tiempo atrás y con la comunicación correcta con el LMS. Le faltaba un detalle, aún no se integraba dentro de los ejercicios interactivos en el modo de evaluación. Se depuraron todos los juegos desarrollados hasta el momento para solventar los fallos posibles que pudieran aparecer frente a todo tipo de casos de utilización.

Se revisaron diseños de diferentes tipos de ejercicios interactivos que para que en la micro-iteración siguiente se pudiera llevar a cabo su desarrollo. Los diseños que se revisaron fueron los siguientes:

- Diseño revisado de un ejercicio interactivo de hardware: Mapas de Karnaugh
- Diseño revisado de un ejercicio interactivo de software: Recorrido de árboles binarios

Se volvió a realizar un repaso importante a los documentos de diseño de todos los ejercicios interactivos y a iterar una nueva revisión del cargador de la consola Nintendo DS.

- Semana VIII

Para esta semana se tuvieron listos dos prototipos de dos ejercicios interactivos de distintos campos, uno de software y otro de hardware, que fueron el 'Recorrido de Árboles' y los 'Mapas de Karnaugh', respectivamente.

Se siguió trabajando en la conexión de la Nintendo DS con el LMS, hubo que revisar el login porque aún tenía algunos problemas de estabilidad de sesión por resolver.

Ahora, cuando un alumno se conseguía identificar, se descargaban del LMS a la consola la lista de cursos matriculados, mostrándolos por pantalla. Por último, se realizaron pruebas rápidas de WiFi DS a través de PALib, la librería de desarrollo para Nintendo DS.

- Semana IX

Para esta semana se tenía preparado un primer borrador de la memoria con algo más de calidad que el que fue entregado en la primera iteración.

Se hicieron versiones finales para los dos tipos de juegos, el de software y el de hardware, para la preparación previa a la entrega final de esta segunda iteración.

- Semana X

Se realizó una demostración a los profesores directores del proyecto sobre los avances en esta segunda iteración. De la misma manera, se entregó una versión de la memoria final para la segunda iteración.

Tercera Iteración

- Semana I

Llegados a estas alturas del curso, la aplicación para la Nintendo DS ya empezaba a tener un tamaño considerable para ser una ROM de esta consola, por lo tanto, al ser una aplicación denominada homebrew, algunos cartuchos utilizados para la carga de las ROM no ejecutaban bien CS-Training, así que hubo que arreglar el problema del tamaño.

En lo referente a Moodle, en los test se tuvo que empezar a colocar campos especiales en algunas de las tablas para tener la información que se necesitaba en las evaluaciones.

- Semana II

Volviendo un poco la vista a atrás, el objetivo era tener 10 tipos de ejercicio interactivo con sus respectivas instancias al finalizar el proyecto, por lo que se empezaron revisar y a documentar la lógica de evaluación de cada ejercicio interactivo.

Había que crear un informe para el instructor, que le ayudara en el uso de Moodle y la creación de los test que se le pondrían al alumno. Por lo tanto, se define y documenta la cadena del campo (marca, número, tipo de cadena, identificadores).

En una nueva revisión del cargador, la pantalla de selección de test debía ser más amigable y visualmente más atractiva. Otro aspecto a tener en cuenta en este apartado es que se debía empezar a preparar una demo con 2 test en un curso para la entrega final.

Se volvió a revisar la memoria de una manera mucho más exhaustiva de lo que se había hecho hasta ahora.

- Semana III

Ya estudiados los diseños de los ejercicios interactivos que se iban a implementar, se pasó a la creación de los prototipos de dichos ejercicios interactivos, dos del campo del software y tres del campo del hardware:

- Ejercicio Interactivo de software: Polimorfismo y herencia
- Ejercicio Interactivo de hardware: Cálculo de intensidades

- Ejercicio Interactivo de hardware: Recorrido de circuitos sencillos
- Ejercicio Interactivo de software: Visibilidad de variables
- Ejercicio Interactivo de hardware: Piezas Lógicas

- Semana IV

Se prepararon y terminaron todos los ejercicios interactivos de cara a poder realizar la entrega final del producto, por lo que se cuidaron los detalles gráficos y de funcionalidad.

- Semana V

Se realizó una entrega de todo el producto final de software, es decir, la Nintendo DS y el Moodle funcionando. Una vez configurados unos test en Moodle, estos se comunicaron a la Nintendo DS para que el alumno los realizara y enviara los resultados de vuelta a Moodle.

La demostración realizada a los profesores directores fue un éxito al no producirse ningún tipo de error ni incidencia.

- Semana VI

Una vez acabada la aplicación en sí, sólo quedaba hacer una memoria de gran calidad que reflejara todo lo que se había aprendido, investigado y desarrollado durante el curso. Para ello se impusieron las fechas que se citan a continuación con el evento correspondiente:

- Entrega del borrador completo: Viernes 12 de junio de 2009
- Entrega del acta: Lunes 22 de junio de 2009
- Entrega de la memoria final en la Facultad: Viernes 3 de julio de 2009
- Defensa pública: 9 de julio de 2009

8. Conclusiones y Trabajo Futuro

En la Introducción de este documento expresábamos nuestro deseo de contar con una propuesta firme e innovadora en el ámbito del e-learning. Llegados a este punto, debemos preguntarnos cuál es la contribución y los beneficios que puede tener el haber desarrollado un modelo de videojuego educativo para una videoconsola portátil en consonancia con un LMS. Se han sobrepasado barreras tecnológicas que impedían comunicar dos plataformas tan distintas como la Nintendo DS y Moodle, sin que ninguna vea sus características limitadas por la otra. A partir de ahora, el poder conjugar los aspectos más importantes de las clases teóricas y presenciales y la enseñanza a través de tecnologías portátiles estará un paso más cerca. Aplicaciones para el resto de carreras universitarias, academias, o colegios, junto a un sistema de gestión del aprendizaje en cada aula, pueden ser realidades en un futuro cercano, partiendo de la base que CS-Training propone. Pocas son las limitaciones de un modelo que puede ser el primer acercamiento a una enseñanza mixta en las aulas, con el uso de las Tecnologías de la Información y Comunicación, en universidades de todo el mundo.

Como una continuación lógica del trabajo elaborado en este proyecto, se abren dos posibles caminos interrelacionados: el desarrollo de una herramienta de autoría, para poder generar automáticamente nuevos contenidos para CS-Training, y el almacenamiento de los ejercicios en la base de datos de Moodle, en un servicio por demanda que permitiría la personalización extrema de los contenidos con los que cada alumno cuenta en su videoconsola.

Herramienta de Autoría

Durante todo el tiempo que ha abarcado el desarrollo de la aplicación se tuvo en mente la idea de poder crear de una manera automática y sencilla las instancias específicas de cada modelo de ejercicio interactivo. Habiendo definido el modelo conceptual de cada tipo de ejercicio, la automatización del proceso de generación de las instancias debería ser algo sencillo y directo. De esta forma, las limitaciones impuestas por la necesidad de mostrar ejemplos de ejercicios incluidos con la aplicación se verían sobrepasadas, con futuras ampliaciones al alcance de cualquiera que deseara contribuir.

Queremos mostrar las facilidades que una herramienta de autoría para los ejercicios interactivos supondría de cara a ampliar la experiencia de juego que ofrece CS-Training, para lo que comentaremos, paso por paso, la creación de un nuevo tipo de instancia a partir de un modelo original.

Creación de una Nueva Instancia con la Herramienta de Autoría

Vamos a dar un ejemplo de diseño de una nueva instancia del ejercicio "Puzzle de Programación", y cómo a partir del modelo general podemos derivar un ejercicio basado en el bucle FOR, una sentencia de control muy común en la programación de casi todos los lenguajes, siendo éste un proceso verdaderamente sencillo.

- El primer paso consiste en identificar el modelo de ejercicio interactivo en el que queremos basarnos. En este caso, el modelo general del ejercicio "Puzzle de Programación", explicado con anterioridad.
- A continuación, identificaremos los elementos de control e interacción comunes a este modelo, como las piezas del puzzle a generar o los botones de reinicio y corrección.
- El siguiente paso consistirá en acotar el problema a desarrollar a partir del enunciado deseado, poniendo un límite al número de piezas-contenedores que habrá en el nuevo ejercicio.
- Por último, bastará con determinar las posiciones finales teóricas que deberán tener cada una de las piezas, y adaptar la lógica de evaluación a éstas, en un proceso directo.
- Ya tendríamos nuestra nueva instancia creada y preparada para añadir al cargador con su evaluador local incluido, y generado en el paso anterior.

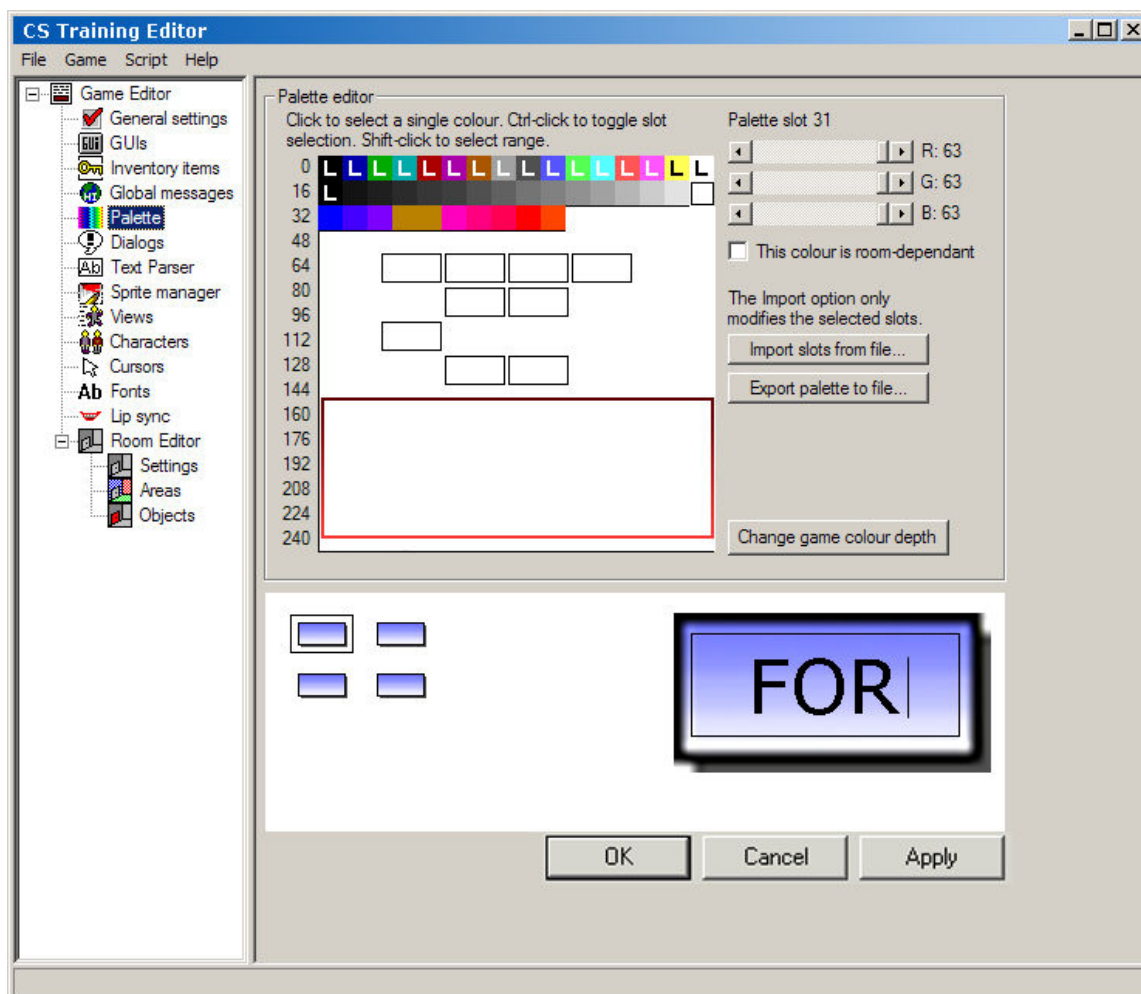


Figura 8.1 Posible captura de la herramienta de autoría

Todos los pasos que se han dado para la creación del nuevo ejercicio basado en el FOR serían idénticos en el caso de querer crear cualquier otra nueva instancia de cualquier otro ejercicio interactivo incluido en CS-Training. La modularidad y escalabilidad de esta aplicación permitiría aumentar la complejidad y el número de contenidos educativos en CS-Training de forma fácil, y con grandes posibilidades. Un ejemplo más de por qué estamos tratando con objetos educativos, siguiendo su patrón de diseño, la capacidad para la generación de nuevos contenidos y la reutilización es significativa.

Ejercicios Interactivos en Moodle

En una reciente ampliación de las librerías PALib (las herramientas utilizadas para desarrollar CS-Training) se introdujo la capacidad de cargar aplicaciones externas desde una determinada. Cuando en un principio se ideó el modelo de CS-Training se tuvo en cuenta la posibilidad de que el contenido que llegaba al cargador, es decir, los ejercicios interactivos junto con su evaluador local, residieran en el LMS.

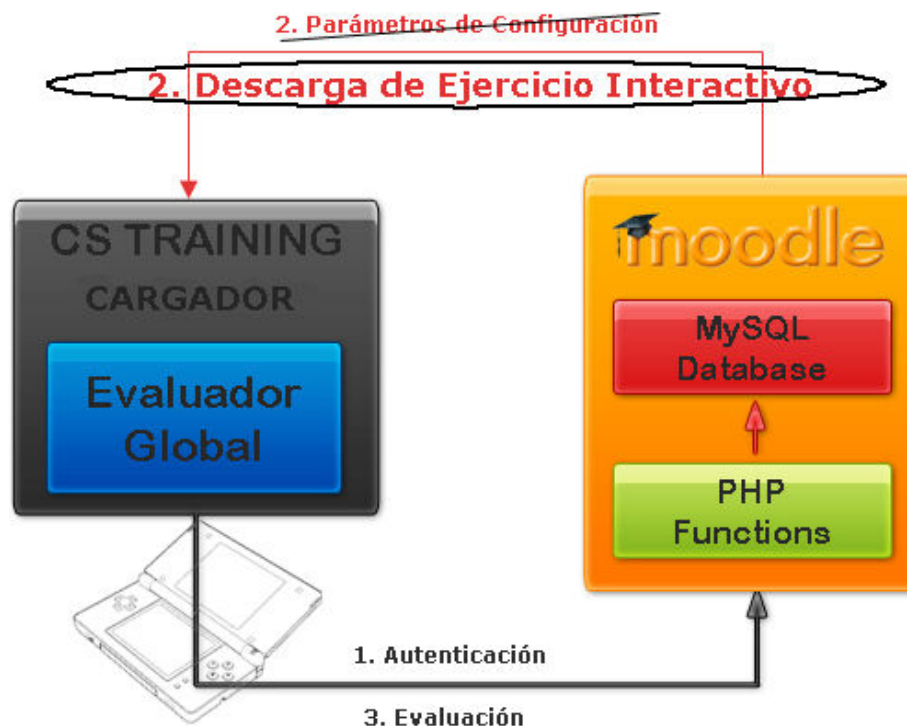


Figura 8.2 Esquema de comunicación con envío de ejercicios interactivos desde Moodle

De esta forma, y en relación con la herramienta de autoría, los contenidos serían personalizados por cada una de las aplicaciones distribuidas entre los alumnos, al nivel de contar con ejercicios específicos por demanda de cada alumno, o instructor. Desgraciadamente, no se pudo llevar a cabo, ya que en su momento no se daba soporte a este tipo de carga de aplicaciones externas. Pero sin duda sería una labor futura de investigación muy interesante el continuar por este camino, para poder ampliar la experiencia educativa de CS-Training a todos los niveles de personalización, sin los límites impuestos por una aplicación cerrada.

Partiendo de la herramienta de autoría, cualquier persona que así lo deseara podría generar contenido en forma de nuevas instancias de ejercicios interactivos y almacenarlas en un repositorio al efecto localizado en Moodle, aumentando considerablemente las posibilidades del modelo de CS-Training.

9. Referencias

- Uso de estándares aplicados a TIC en Educación (CNICE) – Informe 16
- Documentación referida a Moodle: <http://docs.moodle.org/es/Portada>
- Informes del MEC, Videojuegos y Educación
http://ares.cnice.mec.es/informes/02/documentos/iv04_0304c.htm
- Wiki PALib: <http://www.palib.info/wiki/doku.php>
- Manuales de C: <http://www.lawebdelprogramador.com>
- Página web especializada: <http://www.elotrolado.net/>

10. APENDICES

APENDICE A

Librerías PAlib

Las librerías de código abierto PAlib están basadas en las famosas LibNDS de bajo nivel y permiten el desarrollo de aplicaciones de gran envergadura dadas sus semejanzas con el lenguaje C. A lo largo de esta primera parte del apéndice A explicaremos las características y funciones más relevantes que se han usado en el desarrollo de CS-Training, organizadas según su tipo.

Textos

Funciones importantes

Inicialización: *PA_InitText (bool screen, u8 bg_select)*, el primer argumento es la pantalla (1 = superior, 0 = inferior) y el segundo es el fondo (0-3)

Texto simple: *PA_OutputSimpleText (bool screen, u16 x, u16 y, const char *text)*, el argumento screen es la pantalla a usar. x es la coordenada x, dada en tiles (0-31). y es la coordenada y, dada en tiles (0-23)

Texto en una caja: *PA_BoxText (u8 screen, u16 basex, u16 basey, u16 maxx, u16 maxy, const char *text, s32 limit)*, sus parámetros son los siguientes:

screen - la pantalla a usar (0 ó 1)

basex - 'x' de la esquina superior izquierda del área, en tiles

basey - 'y' de la esquina superior izquierda del área, en tiles

maxx - 'x' de la esquina inferior derecha del área, en tiles

maxy - 'y' de la esquina inferior derecha del área, en tiles

text - texto a visualizar

limit - máximo de letras a mostrar en este tiempo (puede usarse también para escritura lenta del texto)

Texto avanzado: *PA_OutputText (bool screen, u16 x, u16 y, const char *text, arguments...)*, es parecida a la función de texto simple, pero en este caso admite argumentos

Color del Texto: *PA_SetTextCol (screen, r, g, b)*, primer parámetro es la pantalla y los tres siguientes corresponden a los colores rojo, verde y azul

Uso

Para utilizar estas funciones, lo más importante es haber inicializado el módulo de texto de la DS, con la función de inicialización en la pantalla en la que queramos mostrar el texto, y usar las funciones de texto sobre la pantalla donde hemos inicializado dicho módulo, de lo contrario no saldrá nada por pantalla

Problemas

El único problema que se puede plantear a la hora de usar estas funciones es que no corresponda la inicialización con el uso del texto en alguna pantalla

Como un problema secundario es donde situar el texto de una manera más personalizada, es decir, solo disponemos de 32 líneas por pantalla.

Uso del Pad

Funciones importantes

No existen funciones como tal, ya que se trata de una estructura llamada Pad y contiene múltiples funcionalidades dependiendo del tipo de pulsación:

Held es por defecto 0, pasa a 1 cuando la tecla se mantiene pulsada.

Released pasa a 1 cuando se deja de pulsar (antes se estaba pulsando y se deja de pulsar), permanece a 1 sólo durante un frame.

Newpress es cuando simplemente pulsas la tecla, permanece a 1 sólo durante un frame.

Uso

Para usar esta estructura solo hay que invocarla para saber su estado

Problemas

No da problemas

Uso del Stylus

Funciones importantes

El Stylus funciona de forma idéntica al pad, es automáticamente actualizado en cada frame y usa una estructura llamada Stylus (simple). Las diferentes variables de Stylus son:

Held, Released, y Newpress, igual que las del Pad

X e Y, usadas para la posición del Stylus

Uso

Para usar esta estructura solo hay que invocarla para saber su estado

Problemas

No da problemas

Uso del teclado

Funciones importantes

PA_InitKeyboard(n): Carga el teclado en el background n

PA_KeyboardIn(X, Y): coloca el teclado en su sitio

PA_SetBgPalCol(Screen, BG, PA_RGB(31,31,31)): Es una función para cambiar el color de fondo. La pantalla debe estar entre 0 o 1 (0 para la pantalla inferior), BG es para el número del background (el del teclado) (0-3) y PA_RGB es el color (0-31). Este no es usado por el teclado persé, pero es la mejor forma para cambiar el color del background.

PA_SetKeyboardColor(0, 1): Es una función para cambiar el color del teclado

PA_CheckKeyboard(): Es un test para saber si has pulsado una tecla con el stylus, y devuelve la palabra que se está pulsando, o 0 si no está pulsado

Uso

Primero se inicializa el texto

Se inicializa el teclado

Se configura con el color correspondiente

Se obtiene la tecla pulsada

Problemas

Estas funciones no dan problemas

Mostrar y desplazar imágenes

Funciones importantes

PA_LoadSpritePal(Pantalla, Num_paleta, Nom_Palet(void)sprite0_Pal)*: carga la paleta del sprite o imagen a mostrar

PA_CreateSprite(Pantalla, Numero de sprite, Nombre de sprite, Tamaño de sprite, Modo, Número de paleta, X, Y): aquí se crea y muestra la imagen

PA_MoveSprite(Num Sprite): comprueba si se está tocando un sprite, y si es así lo mueve. Bastante útil si hay varios sprites

PA_SetSpriteXY(pantalla, sprite, x, y): establece el sprite en una cierta posición

Uso

Primero hay que convertir la imagen en el formato que admite la NDS con las herramientas que se proporcionan con PaLib

Se carga con las funciones especificadas más arriba

Y si se quiere mover, se usa las funciones correspondientes

Problemas

No da problemas siempre y cuando el tamaño establecido por el Sprite sea también su tamaño en realidad

Rotación y zoom de imágenes

Funciones importantes

PA_SetSpriteRotEnable(pantalla, sprite, rotset): habilita la rotación en una imagen

PA_SetSpriteRotDisable(pantalla, sprite): solo escala la imagen

PA_SetRotsetNoZoom(pantalla, rotset, angulo): solo rota la imagen

PA_SetRotset(pantalla, rotset, angulo, zoomx, zoomy): permite rotar y escalar a la vez

Uso

En primer lugar, se crea el sprite como siempre

Después se necesita activar el sprite en el modo rotación/escalado

Asignar un rotset (de 0 a 31), que se hace con *PA_SetSpriteRotEnable* (pantalla, sprite, rotset).

Una vez hecho esto, el sprite está listo para ser rotado, cuando se quiera dejar de rotarlo, se puede usar *PA_SetSpriteRotDisable* (pantalla, sprite).

Problemas

El sprite es rotado desde su punto central

Transparencias y Profundidad en las pantallas

Funciones importantes

PA_EasyBgLoad (pantalla, número de fondo, nombre de fondo):

Uso

Únicamente hay que invocar la función con los parámetros adecuados

Problemas

Si la conversión se hace bien, no existe ningún problema

Funciones matemáticas importantes para juegos

Funciones importantes

PA_InitRand()

PA_Rand()

PA_RandMax(max)

PA_RandMinMax(min,max)

Uso

Se invoca la función de inicialización

Se usa la función que nos convenga mas para la generación de un número aleatorio

Problemas

No suele dar problemas

Programación en 3D

Funciones importantes

Usa las mismas funciones que OPENGL

Uso

El uso es muy similar al de OpenGL sobre plataformas Windows o Linux, pero tiene ciertas sutilezas, como por ejemplo: sólo se pueden dibujar triángulos y cuadrados, y de ahí, el programador debe de crearse otras figuras.

Problemas

A la hora de usarlo en este proyecto, parecía una buena idea, pero a medida que queríamos hacer algo un poco complicado no encontrábamos una manera óptima de hacerlo.

Comunicación

Funciones importantes

```
PA_InitWifi(); //Initializes the wifi
PA_ConnectWifiWFC()
PA_InitSocket(&sock, "www.google.es", 80, PA_NORMAL_TCP)
PA_InitServer(&sock, 9999, PA_NORMAL_TCP, 10)
accept(sock, NULL, NULL)
recv(sock, buffer, 256, 0)
send(sock, buffer, 256, 0)
PA_GetHTTP(buffer, "http://www.invisionsoft.co.uk/ip.php")
Wifi_DisconnectAP()
Wifi_DisableWifi();
```

Uso

El primer paso es inicializar el módulo WiFi con las funciones destinadas a tal fin.

Seguidamente, según el objetivo que se quiera, se usarán unas funciones de transmisión o recepción de datos

Problemas

No es posible configurar los parámetros de Wifi, es decir, ni el punto de acceso al que queremos conectarnos ni el tipo de seguridad, etc. Todo ello hay que realizarlo en un principio con aplicaciones externas.

La Integración con Moodle

Moodle es un sistema de gestión de cursos, de distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea. En términos de arquitectura, Moodle es una aplicación web que se ejecuta en sistemas que soportan PHP, incluyendo la mayoría de proveedores de hosting web. Los datos son almacenados en una sola base de datos MySQL, dividida en los respectivos módulos de los que se compone Moodle (palabra que proviene de "modular") . La versión en el momento del desarrollo de nuestra aplicación es la 1.9, por lo que contaremos con los módulos que trae por defecto el sistema en esta versión.

Configuración de Moodle

La configuración de Moodle que usamos en una primera aproximación es la estándar, sin cambios más allá del estilo visual y personalización de los cursos. La estructura que definiremos para nuestro sistema sigue el esquema de organización representado en la figura siguiente:

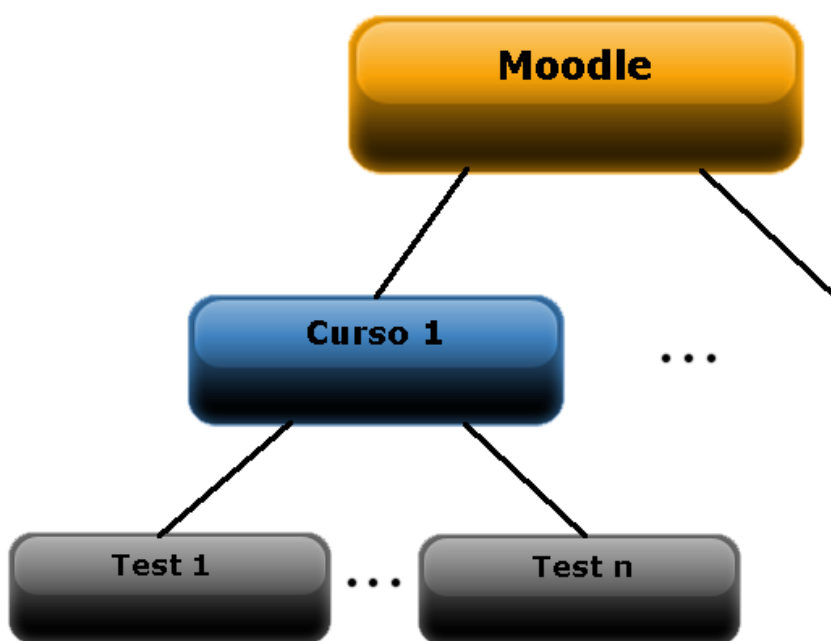


Figura 10.1 Esquema de organización de Moodle

Moodle estará organizado en los distintos Cursos que el instructor desee habilitar, siendo algunos compatibles (con Tests para su realización) con la aplicación de Nintendo DS, y otros no. En estos último el instructor podrá incluir todo el material relacionado con su asignatura que desee, a la vista de los alumnos matriculados. Puede ser un refuerzo de cara a las prácticas que se hagan en el videojuego.

Los cursos que sí sean compatibles contendrán una serie de Tests (creados a partir del módulo Quiz de Moodle) configurados previamente para su realización en NDS. Contendrán

instrucciones básicas sobre cómo realizarlos para cualquier duda que pueda surgirle a un alumno acerca de ello, y una cadena oculta, sólo visible por el instructor, para configurar el contenido, esto es, los ejercicios interactivos de que se compone. Para cada test se guarda una relación de intentos realizados por cada alumno, dispuesta en forma de historial, y resaltando la mayor puntuación obtenida en él hasta la fecha. Cada intento contendrá un informe compuesto por los ejercicios interactivos realizados, la nota obtenida en cada uno de ellos (no la ponderación), y el tiempo necesitado en su resolución.

The screenshot shows the Moodle course configuration page for 'CursoN1'. At the top, it indicates the user is logged in as 'Admin Usuario' and provides a 'Salir' link. The header features a graphic of a laptop with books on it against a green binary background. Below the header, there's a navigation bar with 'NDSc' and a 'Switch role to...' dropdown. The main content area is divided into three columns. The left column has a sidebar menu with sections: 'Personas' (containing 'Participantes'), 'Actividades' (containing 'Cuestionarios' and 'Foros'), 'Search Forums' (with a search box and 'Go' button), and 'Administración' (containing 'Activar edición', 'Configuración', and 'Assign roles'). The middle column shows the 'Diagrama semanal' for the week of 15 de December - 21 de December, with a 'Saltar a...' dropdown and a list of activities: 'Foro de Noticias', 'Prueba', and 'TestMixto'. The right column contains three widgets: 'Noticias' (with a message to add a new topic), 'Upcoming Events' (stating there are no upcoming events), and 'Actividad reciente' (showing activity from Monday, 29 de June de 2009, 03:38).

Figura 10.2 Captura de la configuración de un curso en Moodle

Al contar con un informe detallado, el instructor puede observar con mayor detenimiento (frente al historial de intentos realizado) los avances y progresos del alumno en los conceptos que se hayan practicado. Notas bajas, tiempos de resolución demasiado altos, darán pistas acerca del estado del alumno en la materia, algo muy útil en vista del seguimiento que deben llevar los instructores habitualmente.

Aunque se hayan mostrado los beneficios de contar con tests de varias calificaciones e intentos, el coste de haber encontrado la estructura adecuada para almacenar progresos externos al propio sistema ha sido bastante alto. Moodle está más enfocado a una centralización de todos sus servicios en el propio sistema que a recibir eventos externos y darles soporte, aunque sea algo factible. También se tuvo en cuenta la posibilidad de usar el estándar SCORM para crear unidades educativas que Moodle intercambiara con otros sistemas compatibles y la Nintendo DS, pero su

incompatibilidad con las librerías usadas para el desarrollo de la aplicación (PALib) dieron al traste con esta posibilidad.

Comunicación con Moodle

Las siguientes líneas tratan de explicar de un modo aproximado la estructura de la base de datos que interviene en el almacenamiento del módulo Quiz de Moodle, el que representa nuestros tests evaluados, y cómo se acceden y modifican sus valores.

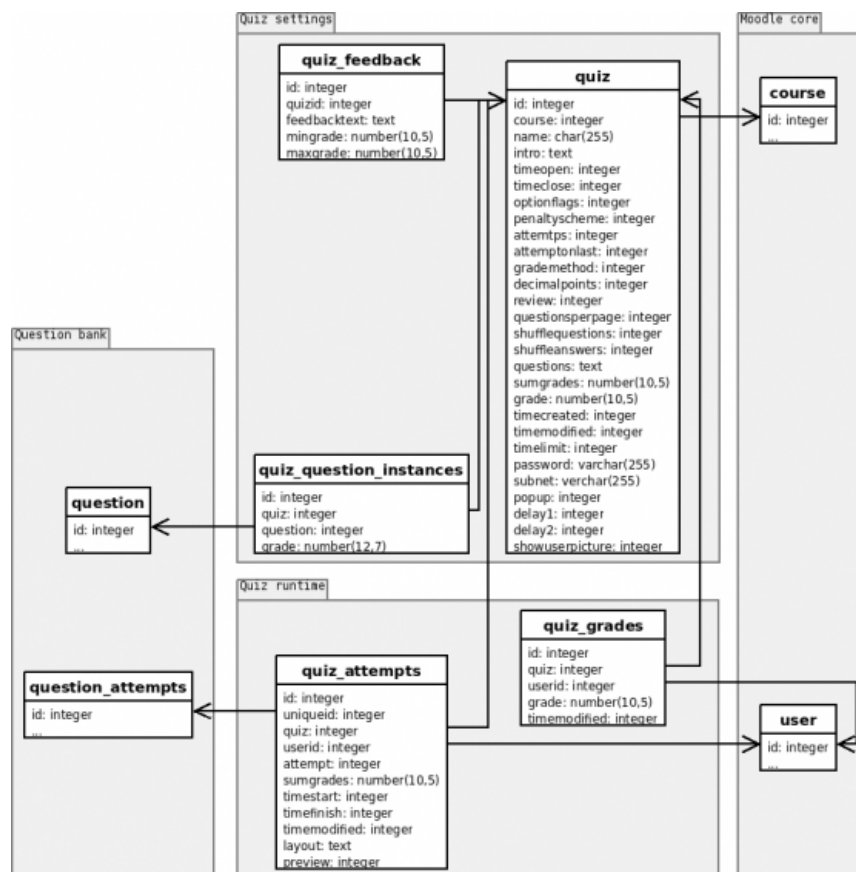


Figura 10.3 Estructura de las tablas de configuración de tests en Moodle

Para poder comunicar a Moodle la nota obtenida por CS-Training ejecutado en Nintendo DS, se hace uso de distintos scripts PHP alojados en el servidor, que mediante consultas MySQL encadenadas modifican los parámetros necesarios de las tablas relacionadas con "Quiz". Las tablas más importantes son las referidas a los intentos (attempts) y al test (quiz). A cada ejecución del modo evaluación de la aplicación, se genera un nuevo intento exclusivo para el alumno en cuestión.

nivel no estaban totalmente depuradas y su funcionamiento no era el deseado, por lo que no era posible establecer y transmitir por una conexión de otro modo.

Frente a todos los problemas de diseño encontrados con el uso de estas librerías gratuitas, lo cierto es que el resultado final es muy satisfactorio, ya que se han cumplido los objetivos previstos en un principio haciendo uso de los únicos métodos al alcance de este desarrollo.

Otras Librerías

Este último punto ha dejado constancia de las dificultades encontradas en el desarrollo de CS-Training, principalmente en el uso de las librerías PALib y sus limitaciones en el momento de comenzar el desarrollo. Por ello se pensó en usar algún otro tipo de librerías de desarrollo que cubriera un mayor espectro de funcionalidades. Lamentablemente, PALib es hoy por hoy la librería más funcional a alto nivel de las que existen, y cualquier cambio habría supuesto una pérdida de soporte y actualización importante.

Creemos firmemente que con el apoyo de la comunidad de desarrolladores que posee PALib se sucederán un sinnúmero de actualizaciones y mejoras con el objetivo de obtener la mejor y más completa librería de desarrollo no oficial para Nintendo DS. Con ello además, CS-Training seguirá siendo compatible y ampliable en un futuro no muy lejano.

APENDICE B

Manual del Profesor

Con la integración de los instructores en el modelo de CS-Training, se requiere una explicación detallada de las funciones extraordinarias que deberán desempeñar, junto con las labores habituales de evaluación, seguimiento, etc. En los dos puntos siguientes, se explican con detalle las características de la cadena de configuración de los tests evaluados, que el instructor determinará desde Moodle a su antojo, y la lógica de evaluación que siguen los ejercicios interactivos, que proporciona una visión sobre la calidad de las respuestas de los alumnos, más allá del simple número en que se ven representadas.

Cadena de configuración de examen

Estructura general de la cadena de configuración:

```
<!--@@N1_X_N2@@-->
```

Como se puede ver, la cadena esta comprendida entre marcas de comentario de HTML. Esto se realiza para que sea transparente para los alumnos y para los profesores en Moodle. A continuación, se procede a explicar cada campo. El campo principal y del que dependen el resto de campos es X, el cual puede tomar los siguientes valores:

'p'

Quiere decir que se configurará para un tipo específico de perfil. Los otros campos quieren decir lo siguiente:

N1: identifica el tipo de perfil con los siguientes valores:

0: Perfil Hardware

1: Perfil Software

2: Perfil Mixto

N2: Esta compuesto por una secuencia de números separados por comas. Los números, idealmente, deben ser del perfil que le corresponde, pero suponiendo que pueden existir errores, si se diera este caso, el ejercicio indicando se ignoraría o se tomaría otro número que correspondiera al perfil.

Un ejemplo de este tipo de secuencias es la siguiente:

```
<!--@@1_p_0,3,4@@-->
```

'i'

Quiere decir que se configurará para un conjunto específico de ejercicios interactivos. Los otros campos quieren decir lo siguiente:

N1: identifica el número de ejercicios que se quiere que aparezcan en el los exámenes

N2: Esta compuesto por una secuencia de números separados por comas. La cantidad de los números, idealmente, debe ser igual o menor que N1. En caso de que sea menor o bien se repiten ejercicios interactivos o sólo se proporcionan los que se indican en la cadena

Un ejemplo de este tipo de secuencias es la siguiente:

```
<!--@@3_i_0,3,4,2@@-->
```

'a'

Quiere decir que se configurará para un conjunto específico de ejercicios interactivos, sin tener el cuenta del perfil de que sean. Los otros campos quieren decir lo siguiente:

N1: identifica el número de ejercicios que se quiere que aparezcan en los exámenes

N2: este campo aparecerá vacío

Un ejemplo de este tipo de secuencias es la siguiente:

```
<!--@@5_a@@-->
```

Lógica de Evaluación

Dada la gran variedad de ejercicios de los que CS Training dispone, basados en asignaturas de la carrera de Ingeniería Informática, existía el reto de encontrar una lógica común para su evaluación. Aún desconociendo la manera en que los profesores encargados de impartir dichas asignaturas corrigen y evalúan a sus alumnos, no es habitual que entre ellos compartan criterios.

De esta forma, en CS Training otorgamos una mayor importancia al número de fallos que el alumno comete, como si de sencillos tests se tratara. Por ello, los ejercicios serán calificados en referencia a la cantidad de errores que se cometan en su realización, siguiendo una determinada plantilla que exponemos a continuación.

Para un ejercicio en el cual la máxima cantidad de fallos que se pueden cometer es de 10:

Ningún fallo: Nota 10.

1 Fallo: Nota 8.

2 Fallos: Nota 6.

3 Fallos: 5.

Más de 3 Fallos: Suspenso, esto es el ejercicio se califica como 0.

Contamos con distintas aproximaciones para cada ejercicio, cambiando su formato pero manteniéndose íntegra la plantilla de corrección.

La nota final se obtendrá haciendo una ponderación de las obtenidas en la realización de estos ejercicios, función de la que se encarga nuestro evaluador global.

FIN