

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA UNIVERSITARIA DE INGENIERÍA TÉCNICA DE
TELECOMUNICACIÓN



PROYECTO DE FIN DE CARRERA

**Desarrollo de herramientas de autoaprendizaje
y evaluación en Señales y Sistemas.**

Carlos Nevado Fernández

Carlos Guardado Ruano

Junio 2016



ETSIS TELECOMUNICACIÓN

PROYECTO FIN DE CARRERA PLAN 2000

TEMA: Señales y Sistemas.

TÍTULO: Desarrollo de herramientas automáticas de aprendizaje aplicadas a Señales y Sistemas.

AUTOR: CARLOS NEVADO FERNÁNDEZ / CARLOS GUARDADO RUANO

TUTOR: JUAN CARLOS GONZÁLEZ SANDE **Vº Bº.**

DEPARTAMENTO: TEORÍA DE LA SEÑAL Y COMUNICACIONES

Miembros del Tribunal Calificador:

PRESIDENTE: JUAN JOSÉ GÓMEZ ALFAGEME

VOCAL: JUAN CARLOS GONZÁLEZ SANDE

VOCAL SECRETARIO: JUANA GUTIÉRREZ ARRIOLA

DIRECTOR:

Fecha de lectura:

Calificación: **El Secretario,**

RESUMEN DEL PROYECTO:

Se trata de implementar algoritmos para la generación y corrección automática de problemas de Señales y Sistemas y su incorporación en la plataforma de Moodle, para facilitar el aprendizaje de los alumnos y mejorar el contacto del profesor con las tareas realizadas por sus alumnos fuera de las aulas.

La implementación de dichos algoritmos se harán en lenguaje Java, ya que este lenguaje nos proporciona una buena solución gráfica para mostrar los ejercicios.

Agradecimientos

Carlos Nevado

Mucha gente ha sido partícipe, directa o indirectamente, para que me encuentre en este momento de mi vida, a todos ellos les debo mucho más que un agradecimiento.

A mis padres que me lo han dado todo, y nunca se han rendido de seguir insistiendo para que terminase esta etapa de mi vida.

A mis hermanos por estar ahí en los momentos de Física, Java, o simplemente estar ahí aunque estén a 400km.

A mis compañeros, en especial al Calvo y a Nachete, sin ellos todos esos momentos de biblioteca, reina y en general horas en la facultad dedicadas a llegar a este punto, no habría sido posible.

A mis amigos y sus ánimos para que no deseperase en la elaboración de este Proyecto.

A Esther, a ti especialmente por aguantarme y siempre animarme y apoyarme cuando sólo vagueaba por las esquinas de casa.

A Juan Carlos de Sande por aguantar nuestras desapariciones durante estos años y darnos la posibilidad de hacer este PFC.

Carlos Guardado

Llegado este momento, debo agradecer a mi familia todo el apoyo mostrado y su gran insistencia para poder terminar este largo camino.

Especialmente mis hijos, que aunque no hayan sido un gran apoyo, si que han sido el motor por el cual terminar esta carrera.

A Elena, por aguantarme tantos años y darme los motores de mi vida.

A mis grandes amigos que siempre han estado apoyándome y dando consejos para poder continuar y sacar adelante este proyecto.

A todos los profesores desde el colegio hasta la universidad, porque entre todos han formado la base para pueda ser lo que soy.

Agradecer a nuestro tutor, la gran paciencia que ha tenido en nuestros altibajos, siempre ha estado dispuesto ayudarnos y facilitarnos las cosas.

RESUMEN

Ya que vivimos en una sociedad cada vez más informatizada y en la cual disponemos de todo tipo de herramientas tecnológicas, nos vemos obligados a intentar sacarles el mayor partido posible. Uno de los puntos que cada vez es más recurrente es el autoaprendizaje haciendo uso de estas herramientas en cualquier momento de tu día a día. Para un alumno de educación superior es una oportunidad más de reforzar sus conocimientos de una asignatura.

La plataforma Moodle, que usan muchas Universidades a nivel mundial, es una plataforma, que facilita el acceso a los alumnos universitarios a información y recursos de las asignaturas que cursan. Unos de los métodos que usa esta plataforma, es compartir el contenido del modelo de referencia de objeto (SCORM). Siguiendo las especificaciones necesarias, hemos usado los SCORM para que dichos alumnos tengan acceso a una aplicación que genera más de 25000 ejercicios diferentes y que se evalúan de forma automática.

El funcionamiento de la aplicación es sencillo. El alumno introduce un número, a modo de semilla, y la aplicación basándose en un algoritmo genera un problema. Cuando el alumno ha terminado de contestar selecciona la opción de corregir y la aplicación evalúa las respuestas del alumno, calculando la solución real del ejercicio y comparándola con las respuestas del alumno. Una vez evaluado el alumno puede volver a realizar el ejercicio o introducir otra semilla para que se le muestre otro problema distinto. Todos los datos introducidos en la aplicación, semilla, respuestas del alumno, etc.; son almacenadas y enviadas al profesor de la asignatura para que este pueda disponer de esa información para la mejor evaluación de su materia.

La aplicación ha sido probada en un grupo de alumnos de la asignatura obligatoria Señales y Sistemas de Ingeniería Eléctrica y Electrónica Grado de la Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación de la Universidad Politécnica de Madrid, y el estudio se muestra en un capítulo de este proyecto final de carrera.

ABSTRACT

Since we live in an increasingly computerised and connected society, where we have all kinds of technologic tools available, we need to know how to take advantage of it. One of the most important concepts nowadays is that of self-learning, especially having these wide range of tools at our disposal at any time. For university students, they are an excellent opportunity to revise and reinforce their learning process.

MOODLE is a learning platform used by most universities and other institutions worldwide. It facilitates the access to information and contents of the subjects the students are learning. One of the resources used by the platform to create contents and exercises is the application SCORM. SCORM is a set of technical standards for e-learning software that we have used to give the students access to more than 25000 self-assessment online exercises.

SCORM has a very easy functioning. The student introduces a number as a seed and the application, using an algorithm, generates an exercise. When the student has finished doing it, selects the option “correction” and the application assesses the answers of the student, comparing them with the actual solution. Once the student has been assessed, the process can be repeated again redoing the exercise or introducing a new seed to generate a new problem. All the data, seeds and answers, are stored and sent to the teacher to be used in the assessment process.

The application has been tested by a group of students of the subject Señales y Sistemas for the degree of Electrical and Electronics Engineering at Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación at Universidad Politécnica de Madrid. This study is shown in one of the chapters at the end of this degree final project.

ÍNDICE

RESUMEN.....	V
ABSTRACT.....	VI
ÍNDICE	7
ÍNDICE DE FIGURAS.....	9
ÍNDICE DE ACRÓNIMOS.....	11
1.INTRODUCCIÓN	13
1.1 Motivaciones	13
1.2 Objetivos	15
1.3 Propuesta de la solución.....	16
1.4 Estructura del documento.....	16
2. MARCO TECNOLÓGICO.....	19
2.1 Applets.	19
2.1.1 Restricciones de los Applets.	20
2.1.2 Ventajas de los applets.....	20
2.2 Modelo de objetos del documento.	20
2.3 SCORM.....	21
2.4 Sistema de Gestión de Aprendizaje.....	21
3. DESCRIPCIÓN DE LA ARQUITECTURA PRINCIPAL.....	23
3.1 Descripción de la arquitectura global.....	23
3.2 Módulo generador de problemas.....	24
3.3 Tareas realizadas	26
3.3.1 Ejercicio discreto.....	26
3.3.1.1 Generación del enunciado	26
3.3.1.2 Almacenar resultado.....	28
3.3.2 Ejercicio continuo.	29
3.3.2.1 Generación del enunciado	30
3.3.2.2 Obtención de la convolución en tiempo continuo.....	32
3.3.4 Ejercicio gráficas.....	35
3.3.4.1 Modificaciones Módulo Enun_Manager.....	35

3.3.4.2 Modificaciones en el Applet	36
3.3.4.2.1 Modificaciones en el bloque Funciones.....	36
3.3.4.3 Módulo figuras	37
4. PRUEBAS Y RESULTADOS	41
4.1 Ejercicio discreto.....	41
4.2 Ejercicio continuo.	42
4.3 Ejercicio gráficas.....	44
4.3.1 Ejercicio discreto con gráficas.	45
4.3.2 Ejercicio continuo con gráficas.....	47
4.4 Almacenamiento de datos.	49
5. ESTUDIO DEL IMPACTO DE LA APLICACIÓN EN ALUMNOS	55
5.1 Ejercicio discreto.....	55
5.2 Ejercicio continuo	57
5.3 Conclusión	59
6. CONCLUSIONES	61
REFERENCIAS	63

ÍNDICE DE FIGURAS

Fig. 2.1 Arquitectura de un sistema con applets.....	18
Fig. 3.1 Arquitectura propuesta.....	23
Fig. 3.2 Arquitectura módulo generador de problemas.....	24
Fig. 3.3 Forma en la que el profesor recibe los datos de la resolución de un ejercicio por parte de un alumno.....	28
Fig. 3.4 Tipo de funciones en tiempo continuo.....	31
Fig. 3.5 Secuencia de coordenadas. Donde 1,2 y3 son las distintas coordenadas que necesitamos almacenar.....	38
Fig. 4.1 Ejercicio discreto corregido el error números múltiples.....	40
Fig. 4.2 Ejercicio continuo corregido el error números múltiples.....	41
Fig. 4.3 Ejercicio continuo comprobación de la convolución para la semilla 30030...42	
Fig. 4.4 Ejercicio continuo comprobación de la convolución para la semilla 58.....	44
Fig. 4.5 Enunciado con ambas funciones de forma analítica.....	44
Fig. 4.6 Enunciado con una función en forma analítica y la otra en forma gráfica....	45
Fig. 4.7 Enunciado con ambas funciones en forma gráfica.....	45
Fig. 4.8 Enunciado con ambas funciones de forma analítica.....	46
Fig. 4.9 Enunciado con una función en forma analítica y la otra en forma gráfica....	46
Fig. 4.10 Enunciado con ambas funciones en forma gráfica.....	47
Fig. 4.11 Ejemplo con semilla 57.....	48
Fig. 4.12 Ejemplo con semilla 12.....	49
Fig. 4.13 Ejemplo con semilla32.....	50
Fig. 4.14 Datos alumno.....	51
Fig. 5.1 Evolución de los alumnos en el ejercicio discreto.....	54
Fig. 5.2 Resultados del alumno2.....	54
Fig. 5.3 Resultados del alumno4.....	55
Fig. 5.4 Evolución de los alumnos en el ejercicio continuo.....	56

Fig. 5.5 Resultados del alumno1.....	56
Fig. 5.6 Resultados del alumno6.....	57
Fig. 5.7 Historia gráfica resultados del examen.....	57

ÍNDICE DE ACRÓNIMOS

API	Application Programming Interface.
DOM	Document Object Model.
GUI	Graphic User Interface.
HTML	HyperText Markup Language.
IEEE	Institute of Electrical and Electronics Engineers.
JVM	Java Virtual Machine.
LMS	Learning Management System.
LTI	Linear Time-Invariant.
SCO	Shareable Content Object.
SCORM	Sharable Content Object Reference Model.
XML	eXtensible Markup Language.

1.INTRODUCCIÓN

1.1 Motivaciones

Nos movemos en la era de la información y la comunicación, hoy en día prácticamente todo el mundo tiene al alcance de su mano la posibilidad de acceder a cualquier tipo de conocimiento, a través de un ordenador, desde su casa, biblioteca, trabajo, centro de estudio, etc. Cualquier persona puede ser autodidacta, o buscar ejercicios y aplicaciones que le ayuden a complementar sus conocimientos.

Esta es una idea que no pasa desapercibida para la educación universitaria europea. Con la implantación del Espacio Europeo de Educación Superior, cuyos objetivos quedan recogidos en la Declaración de Bolonia [UE 1999], la Educación Superior está sufriendo grandes cambios y transformaciones debido a la aplicación de nuevos enfoques educativos, los cuales perciben al estudiante como un agente activo en el proceso de la adquisición del conocimiento.

Estos cambios se notan cada vez más en la forma de enseñar, lo cual se nota en la afloración de nuevos métodos y estructuras que faciliten la canalización de la información a los alumnos, así como intentar hacerles partícipes de forma activa en el aprendizaje. Como ya dijo en su día Benjamín Franklin: “Dime y olvido, enséñame y lo recuerdo, involúcrame y lo aprendo”, esta frase resume muy bien lo que hemos intentado plasmar con este proyecto final de carrera, una involucración a la hora de afianzar los conocimientos adquiridos en las aulas facilitando a los alumnos el acceso interactivo a una gran fuente de ejercicios.

Estos cambios en la metodología docente lógicamente obligan a una modificación de los modos de evaluación. Por ello se deben de desarrollar técnicas de evaluación adaptadas a las nuevas metodologías, así como dotar al alumno de la posibilidad de la autoevaluación como signo de su autonomía a la hora de afianzar conocimientos con este nuevo sistema que precisamente intenta fomentar el trabajo autónomo del alumno.

Como decíamos anteriormente vivimos en una sociedad de la información y la comunicación y la educación debe de adaptarse a ello, y aprovecharse de las ventajas que se le presentan de cara a la transformación de los métodos tradicionales de trabajo. Un claro ejemplo de esta adaptación es el b-learning.

El *b-learning* (formación combinada, del inglés *blended learning*) consiste en un proceso docente semipresencial [Coaten 2003]; esto significa que un curso impartido con este formato incluirá tanto clases presenciales como actividades de *e-learning*. Este modelo de formación hace uso de las ventajas de la formación 100% *on-line* y la formación presencial, combinándolas en un solo tipo de formación que agiliza la labor tanto del formador como del alumno. El diseño instruccional del programa académico para el que se ha decidido adoptar una modalidad *b-learning* deberá incluir tanto actividades on-line como presenciales, pedagógicamente estructuradas, de modo que se facilite lograr el aprendizaje buscado. Las ventajas que se suelen atribuir a esta modalidad de aprendizaje son la unión de las dos modalidades que combina:

- Las que se atribuyen al *e-learning*: la reducción de costes, acarreados habitualmente por el desplazamiento, alojamiento, etc., la eliminación de barreras espaciales y la flexibilidad temporal, ya que para llevar a cabo gran parte de las actividades del curso no es necesario que todos los participantes coincidan en un mismo lugar y tiempo.
- Las de la formación presencial: interacción física, lo cual tiene una incidencia notable en la motivación de los participantes, facilita el establecimiento de vínculos, y ofrece la posibilidad de realizar actividades algo más complicadas de realizar de manera puramente virtual.

El motor principal de un sistema *b-learning* es el sistema de gestión del aprendizaje, el cual, entre otras cosas, es el responsable de la distribución y gestión de acceso a los contenidos de aprendizaje. En la comunidad del *e-learning* este tipo de sistemas es conocido como sistemas de gestión del aprendizaje (LMS del inglés: Learning Management System).

Las plataformas LMS integran herramientas de creación de test y cuestionarios. Estudios realizados han constatado la poca eficiencia de estas plataformas cuando se trata de manejar grandes grupos de alumnos o gran número de autoevaluaciones, así como el inconveniente de que el carácter de preguntas que implementan es básico para el nivel que los estudios de ingeniería demandan.

Además de lo comentado anteriormente es necesario disponer de un amplio banco de preguntas que se ajusten al nivel de los estudios en cuestión.

1.2 Objetivos

Durante el curso docente 2010/2011 la ingeniera técnica en telecomunicaciones Ana Osorio Navarro comenzó el desarrollo de un proyecto final de carrera bajo el título de “Desarrollo de herramientas de autoaprendizaje y evaluación en señales y sistemas” con la realización de dos ejercicios que intentasen dar solución a todos o gran parte de los problemas presentados anteriormente. Con todo ello, nuestra predecesora se planteó los siguientes puntos principales a solucionar:

- ⤴ Generar de manera automática miles de variaciones numéricas diferentes de un conjunto de ejercicios tipo.
- ⤴ Evaluar automáticamente los ejercicios.
- ⤴ Ser flexible a modificaciones y adaptaciones de acuerdo al tipo de ejercicio que se pretenda integrar en el sistema a construir, y sus características particulares, minimizando el esfuerzo y coste y maximizando la reutilización.
- ⤴ Ser integrable en la plataforma de b-learning de la Universidad Politécnica de Madrid (Moodle).
- ⤴ Realizar el seguimiento de las calificaciones del estudiante, mientras interactúa en un sistema b-learning.

Concretamente recibimos un ejercicio de sistemas LTI en tiempo continuo y otro en tiempo discreto. Grosso modo en ambos ejercicios debemos depurar el funcionamiento y mejorar en la medida de lo posible estos ejercicios además de realizar un algoritmo para obtener y proporcionar al Moodle las respuestas dadas por el alumno para así facilitar al profesor el seguimiento de los progresos de los alumnos, y específicamente en el ejercicio de tiempo continuo nos vemos obligados a plantear de nuevo el algoritmo y codificación del cálculo de la convolución ya que no funciona correctamente. Respecto al otro ejercicio que recibimos nuestros esfuerzos se tienen que centrar en conseguir la comunicación de los datos de los alumnos con el Moodle, para su posterior tratamiento por parte del profesor.

Además de esto debemos llevar a cabo la realización de dos ejercicios que aporten una versión gráfica a los ejercicios ya desarrollados.

Con todo esto se nos plantea como proyecto final de carrera dar continuidad a estos ejercicios para que puedan ser usados en la plataforma Moodle por parte de los alumnos para complementar sus conocimientos. Por ello los objetivos que tenemos que intentar cumplir son:

- ⤴ Depuración de errores de los ejercicios existentes.
- ⤴ Mejora de los ejercicios existentes.
- ⤴ Realizar el seguimiento de las calificaciones del estudiante, mientras se interactúa en un sistema b-learning.
- ⤴ Realizar dos ejercicios nuevos.

1.3 Propuesta de la solución.

Teniendo en cuenta que partimos de un trabajo ya iniciado, esto nos da una gran ventaja debido a que no tenemos más que plantearnos ciertos problemas iniciales como la mejor forma de diseño de la aplicación y cosas similares. Así mismo esto nos plantea un problema inicial grande, ya que nuestro primer objetivo es entender y comprender el código fuente de las aplicaciones realizadas por nuestros predecesores para poder afrontar el resto de objetivos con garantías. Tras realizar un primer acercamiento al código fuente y atendiendo las peticiones realizadas por parte de nuestro tutor nos planteamos como soluciones principales:

- ⤴ Realizar una mejora sustancial de algunas partes de los ejercicios.
- ⤴ Conseguir una comunicación de los ejercicios con el sistema LMS para poder realizar un seguimiento de los progresos de los alumnos.
- ⤴ Realizar dos nuevos ejercicios como mejora de los desarrollados anteriormente.

El objetivo principal que se nos presenta es la realización de los dos ejercicios basados en los desarrollados con anterioridad, así como solventar los problemas de comunicación de los ejercicios con la plataforma Moodle para que el profesor tenga a su disposición una información de cara a reforzar sus clases.

1.4 Estructura del documento

El documento del proyecto está estructurado en 7 capítulos:

El capítulo 1 contiene una breve introducción de lo que ha sido el trabajo fin de carrera, detallando las motivaciones que nos impulsan a su desarrollo, haciendo una pequeña presentación del punto de partida así como un primer acercamiento a la solución dada.

El capítulo 2 describe el marco tecnológico en el que nos hemos movido para la realización del trabajo. Debido a que este PFC es una continuación del presentado por Ana Osorio Navarro (“Desarrollo de herramientas de autoaprendizaje y evaluación en Señales y Sistemas”) este apartado será un breve resumen de las tecnologías usadas, como son en primer lugar el concepto de Applet, en segundo lugar se comenta el Document Object Model (DOM), interfaz de programación de aplicaciones (API) para documentos HTML (HyperText Markup Language, o lenguaje de Marcado de Hipertexto) y XML (eXtensible Markup Language, o lenguaje de marcas extensible); en tercer lugar el modelo de referencia de contenidos de *b-learning* SCORM; y por último el concepto de sistemas de gestión del aprendizaje (Learning Management System, LMS).

El capítulo 3 engloba el diseño de las soluciones realizadas para cada uno de los ejercicios. Al igual que en el capítulo anterior como el trabajo se realiza basándonos en unos ejercicios existentes de un PFC anterior se realizará un pequeño resumen de la

solución dada, adentrándonos más a fondo en las modificaciones practicadas por nosotros, como el nuevo cálculo de la convolución, los módulos que permiten la comunicación de datos o el módulo figuras imprescindible para los nuevos ejercicios.

El capítulo 4 nos muestra un estudio pormenorizado del funcionamiento de los dos nuevos ejercicios así como pruebas de su correcto funcionamiento y del aspecto de la información almacenada en Moodle para el posterior uso de ella por parte del profesor.

El capítulo 5 describe un pequeño estudio del impacto del uso de estos ejercicios en la mejora académica de los alumnos y las conclusiones.

El capítulo 6 detalla las conclusiones obtenidas tras la realización de este proyecto final de carrera.

El capítulo 7 contiene las referencias bibliográficas usadas para la realización del proyecto final de carrera y la memoria de este.

2. MARCO TECNOLÓGICO

2.1 Applets.

Un applet es un subprograma que se carga y se ejecuta en un Navegador Web como parte de una página HTML [Ceballos 2005b]. El servidor Web donde se encuentra la página que contiene al applet contiene también los archivos que este necesita para ejecutarse. En navegador descarga estos archivos e inicia la ejecución del applet, la cual se lleva a cabo en su totalidad del lado del cliente.

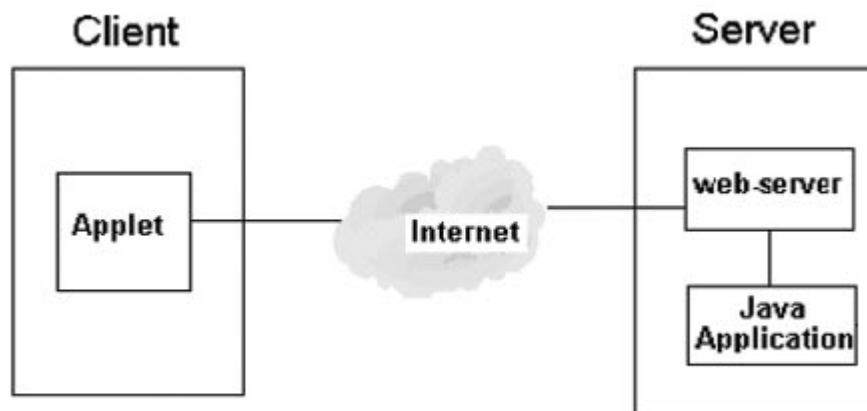


Fig. 2.1 Arquitectura de un sistema con applets.

2.1.1 Restricciones de los Applets.

Dependen del navegador, y a su vez según el navegador dependen de los controles de seguridad.

Principales restricciones de los applets:

1. Un applet no puede leer ni escribir en el disco local.
2. Un applet no puede arrancar ningún programa del ordenador del usuario donde se está ejecutando.
3. Un applet no puede leer ciertas propiedades del sistema.
4. Un applet no puede realizar conexiones de red.

2.1.2 Ventajas de los applets.

Las principales ventajas que tienen son:

1. Son multiplataforma, funcionan en cualquier sistema operativo para el cual exista una JVM (Java Virtual Machine).
2. No necesitan instalación.
3. No hay que preocuparse por la presencia de código que pueda causar daños en el sistema del cliente debido a las restricciones de seguridad y estructura de los applet.
4. Puede trasladar el trabajo del servidor al cliente, haciendo una solución Web más estable tomando en cuenta el número de usuarios/clientes.

2.2 Modelo de objetos del documento.

El Modelo de Objetos de Documento (DOM) es una interfaz de programación para los documentos con formato XML y HTML.

Una interfaz de programación, también conocida como una API (Interfaz de programación de aplicaciones para), es un conjunto de herramientas para la comunicación entre varios programas o, en este caso, lenguajes diferentes. La API se verá a menudo, sin importar el lenguaje de programación que se aprendiendo.

DOM es una API que se utiliza con documentos XML y HTML, lo que nos permitirá, a través de Javascript, acceder a documentos en XML y / o HTML. Es a través del DOM

que podemos cambiar los elementos HTML (pantalla u ocultar un div por ejemplo), añadir, mover o incluso eliminar.

La aplicación a nuestro proyecto de esto, queda reflejada en que es necesaria una API para la comunicación entre un Objeto de Contenido (SCO) y el LMS con un determinado localizador, para lo que usamos el objeto window de DOM.

2.3 SCORM

En su definición más simple SCORM es un modelo que hace referencia a un conjunto de estándares técnicos y especificaciones interrelacionados diseñados para alcanzar requerimientos de alto nivel en el desarrollo de contenidos y sistemas de aprendizaje [ADL 2004b].

SCORM tampoco especifica la tecnología que tenemos que usar, con la excepción del contenido se pueda lanzar en un navegador Web, el contenido se visualiza como html y la comunicación entre el contenido y el LMS requiere Javascript.

Los aspectos fundamentales que cubre SCORM se pueden sintetizar de la siguiente manera:

- Define:
 - Como construir un LMS.
 - Cómo debe ser la estructura de un Objeto Educativo y como empaquetarlo.
 - Cómo y qué información se puede intercambiar entre el LMS y el contenido.
 - Cómo crear contenido adaptativo.
 - Cómo identificar los Objetos Educativos con metadatos.

- Permite:
 - Importar/exportar objetos SCORM entre la herramienta de autoría y el LMS.
 - Intercambiar Objetos Educativos entre LMS.
 - Reusar el contenido de distintos Objetos Educativos.
 - Secuenciar el contenido basado en las interacciones del estudiante.

2.4 Sistema de Gestión de Aprendizaje.

Un LMS o sistema de gestión de Aprendizaje es un software instalado en un servidor que se utiliza para la gestión y distribución de cursos a través de internet. El objetivo fundamental del sistema de gestión de aprendizaje es servir de contenedor de cursos, pero también puede incorporar otras herramientas para facilitar la comunicación y el trabajo colaborativo entre profesores y estudiantes, herramientas de seguimiento y evaluación del alumno, etc.

El LMS puede considerarse como una plataforma o campus virtual, donde alumnos, tutores, profesores o coordinadores se conectan a través de internet para descargar contenido, ver el programa de asignaturas, enviar tareas, contestar cuestionarios, debatir en un foro, etc.

Para SCORM, LMS significa un entorno basado en el servidor que tiene la capacidad de gestionar y liberar contenidos a los estudiantes. Determina qué liberar y cuándo y traza el progreso y la interacción del estudiante con un curso

3. DESCRIPCIÓN DE LA ARQUITECTURA PRINCIPAL

3.1 Descripción de la arquitectura global.

Como ya hemos comentado partimos de un trabajo ya hecho, por lo que mantenemos unas bases a las que tenemos que ceñirnos.

El proyecto que nos encontramos consta de tres grandes módulos principales (Ver figura 3.1):

- Módulo generador de problemas.
- Módulo Gestor.
- Adaptador del LMS (API Adapter).

El módulo generador de problemas está pensado para poder ser ampliable o modificado teniendo en cuenta la necesidad de incluir nuevos ejercicios para los alumnos. Por ello está compuesto por varios paquetes que engloban las diversas funciones que tiene dicho módulo.

Los otros dos módulos son más específicos, con el módulo gestor controlamos la sesión, y el adaptador de LMS es necesario para la comunicación con los sistemas de b-learning descargando el contenido desde un servidor LMS.

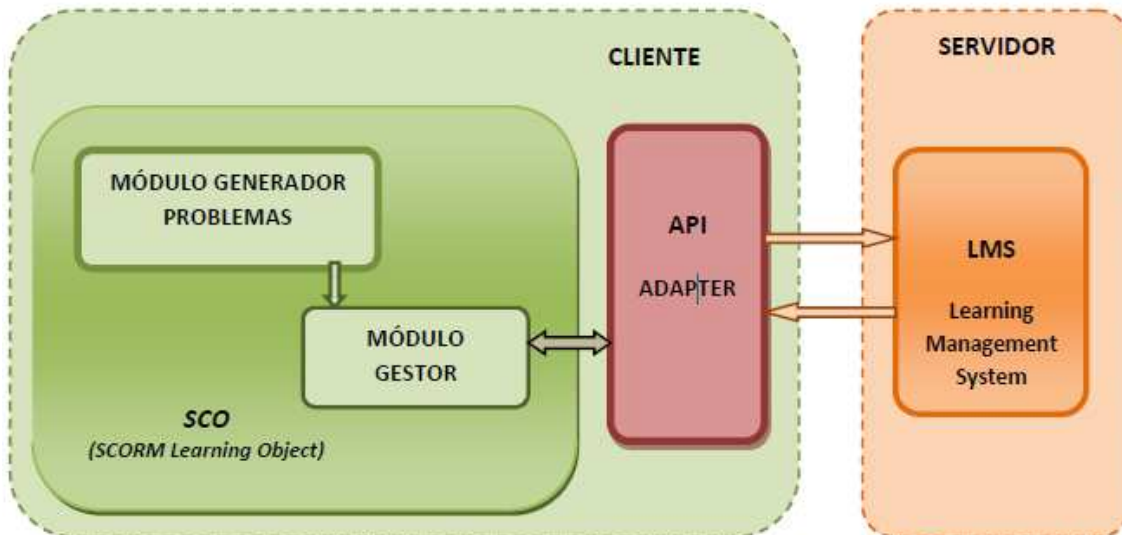


Fig. 3.1 Arquitectura propuesta

El módulo generador de problemas es el núcleo principal del programa. Se genera una interfaz en la que se mostrara el problema al alumno. Partiendo de la base de un número que introduce el alumno (semilla), se calculan unos parámetros para el enunciado. El alumno introduce sus soluciones para el ejercicio planteado. El alumno finaliza el ejercicio pulsando el botón corregir, en ese momento se calculan las soluciones del ejercicio y se comparan con las aportadas por el alumno, obteniéndose así una nota. Estas notas se almacenan y se envían al profesor para así poder llevar un control sobre los alumnos.

El módulo gestor de sesión se encarga de generar una sesión, abrirla y gestionar toda la información que va a ser reenviada al profesor, esto es, almacenar para cada semilla introducida, las respuestas correctas, las introducidas por el alumno y la nota obtenida en cada uno de los intentos. Finalmente se cierra y destruye la sesión.

Para poder comunicar la plataforma de b-learning con nuestra aplicación, es necesario que subamos nuestros datos de sesión desde nuestra aplicación al servidor de LMS integrado en la plataforma de b-learning, de esto se encarga el módulo LMS.

3.2 Módulo generador de problemas

Este es el modulo principal de la aplicación. Partiendo de la base de una semilla introducida por el alumno se generan problemas con diferentes parámetros, para que el alumno pueda practicar con problemas distintos de diferentes dificultades simplemente introduciendo un dato.

Este módulo está compuesto por varios paquetes (ver figura 3.2), cada uno de ellos con una misión concreta e independiente del resto.

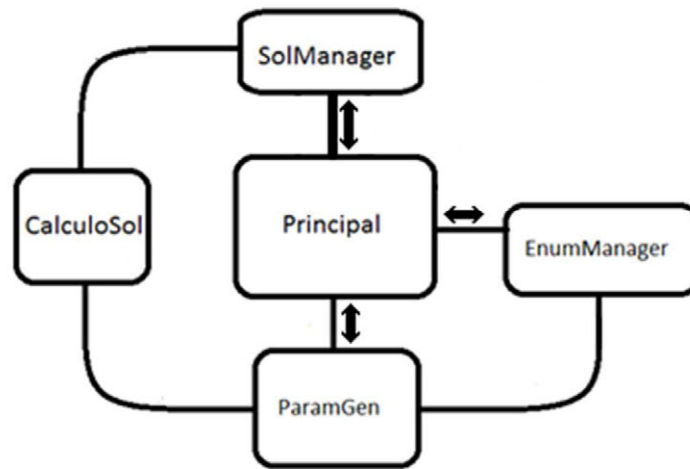


Fig. 3.2 Arquitectura módulo generador de problemas

La función de cada uno de ellos es:

- Principal .- Se encarga del control de la interfaz que maneja el usuario. Es el paquete principal del que dependen el resto de paquetes del módulo generador de problemas.
- ParamGen .- Genera los parámetros de un nuevo problema a partir de una semilla introducida por el usuario.
- EnumManager .- Es el encargado de generar el enunciado del nuevo proyecto a partir de los datos proporcionados por el generador de parámetros.
- SolManager .- Es el encargado de corroborar que las soluciones aportadas por el usuario son correctas basándose en las soluciones obtenidas en el paquete cálculo de la solución.
- CalculoSol .- Este paquete obtiene la solución del problema planteado.

Como se puede apreciar esta estructura basada en paquetes cada uno de ellos con su función nos proporciona gran libertad a la hora de poder modificar o plantearnos la posibilidad de expandir el programa con nuevas funciones.

3.3 Tareas realizadas

En los siguiente sub-apartados vamos a describir las tareas o tipo de ejercicios que hemos desarrollado para el apoyo al alumno para obtener los conocimientos que se evaluarán en la asignatura. Todos estos tipos de ejercicios cumplen con la arquitectura anteriormente descrita.

3.3.1 Ejercicio discreto

Este ejercicio fue heredado del **PFC de Ana Osorio**. Nuestro tutor nos traslada unas observaciones sobre diversos fallos en el funcionamiento del programa.

Los primeros requerimientos son:

- Eliminación de un fallo a la hora de obtener el enunciado.
- La modificación de la forma de calcular la solución de la convolución, para que sea correcta en todos los supuestos.
- El almacenamiento de toda una serie de datos requeridos por el Tutor para poder realizar un mejor seguimiento del aprendizaje de los alumnos.

3.3.1.1 Generación del enunciado

En esta parte del ejercicio se obtiene un enunciado a partir de una semilla introducida por el alumno. El enunciado se compone de dos señales de la forma:

$$x[n] = A(u[n \pm n_1] - u[n - n_2]); \quad (1)$$

$$h[n] = n^N(u[n \pm n_3] - u[n - n_4]); \quad (2)$$

Para calcular los valores de los parámetros n_1 , n_2 , n_3 , n_4 , y N de estas funciones a partir de la semilla se generan una serie de parámetros tales como se detalla a continuación.

Semilla:S.

$$M2 = S\%2; \quad (3)$$

$$M3 = S\%3; \quad (4)$$

$$M5 = S\%5; \quad (5)$$

$$M7 = S\%7; \quad (6)$$

$$Mm2 = (S\%11)\%2; \quad (7)$$

$$Mm3 = (S\%13)\%3; \quad (8)$$

$$Mm5 = (S\%13)\%5; \quad (9)$$

$$Mm7 = (S\%11)\%7; \quad (10)$$

El símbolo % indica el resto entero resultante la dividir por un entero.

Almacenamos los datos en un vector

$$p = (M2, M3, M5, M7, Mm2, Mm3, Mm5, Mm7); \quad (11)$$

Con estos parámetros obtenemos los valores n_1, n_2, n_3, n_4 , y N que aparecen en (1) y (2) de la siguiente forma.

Algoritmo de la obtención del enunciado:

Para obtener $x[n]$

Escribimos “ $x[n]=(1+M3)(u[n]$ ”

Si((3-M7) mayor que 0)

Escribimos “+”

Si((3-M7) distinto de 0)

Escribimos “3-M7”

Escribimos “]-u[n”

Si(la semilla no es múltiplo de los dividendos)

Si ((1-M7-Mm4) es distinto de 0)

Escribimos ”1-M7-Mm4]”

Si no(la semilla si es múltiplo)

Escribimos “+1]”

Obtenemos una función $x[n]$ de la forma:

$$x[n] = (1 + M3)(u[n \pm (3 - M7)] - u[n - (1 - M7 - Mm4)]); \quad (12)$$

Como se puede observar es un caso particular de la ecuación (1).

Para obtener $h[n]$

Escribimos “ $h[n]=$ ”

Si($M2$ es igual a 1)

Escribimos “ n ”

Sino

Escribimos “ n^{M2} ”

Escribimos “ $u[n]$ ”

Si($(2-M5)$ es mayor que 0)

Escribimos “ $+$ ”

Si($(2-M5)$ es distinto de 0)

Escribimos “ $2-M5$ ”

Escribimos “ $]u[n$ ”

Escribimos “ $2-M5-Mm3-3]$ ”

Obtenemos una función $h[n]$ de la forma:

$$h[n] = n^{M2}(u[n \pm 2 - M5] - u[n - 1 - M5 - Mm3]); \quad (13)$$

Como se puede observar es un caso particular de la ecuación (2).

3.3.1.2 Almacenar resultado

El principal objetivo de este PFC, desde un punto de vista puramente académico, es el de facilitar al profesor una visión del estado de su alumnado respecto a la materia impartida sin tener que realizar un examen. Con ello podrá reforzar conceptos o repetir explicaciones según sean los datos obtenidos de la realización de los ejercicios.

Para conseguir esto, se nos pide que traslademos el mayor número de datos de la resolución del ejercicio. Los datos que se envían de cada ejercicio al profesor son los siguientes:

Semilla: Es el número introducido por el alumno a partir del cual se generará el enunciado del problema.


Respuestas alumno: Son las respuestas dadas por el alumno para cada una de las preguntas del ejercicio, independientemente de si estas son o no correctas.

Respuestas correctas: Son las respuestas correctas a cada uno de los apartados del ejercicio.

Notas: Es la obtenida por el alumno al realizar el ejercicio.

Para almacenar y después mandar esta información de los ejercicios hechos por los alumnos desarrollamos una serie de funciones dentro del módulo GestorDeSesion (Ver figura 3.3).

- SetRespuestas: Almacena las respuestas del alumno.
- SetRespuestas_Ok: Almacena las respues correctas del ejercicio.
- SetNotaObj:Almacena la nota obtenia del del alumno en la resolución del ejercicio. La primera nota la guarda como puntuaión del ejercicio. Las siguientes puntuaciones de los siguientes intentos los almacena para mandar al profesor a nivel informativo.
- EnviarInfoAILMS: Gestiona el envío de la información de las pruebas realizadas.

Elemento	Valor
cmi.core.lesson_status	completed
cmi.core.score.raw 	1
cmi.interactions_0.id 	Seed:3 OK:0.0,6.0,6.0,5.0,8.75,10.0 Seed:7 OK:-2.0,5.0,7.0,0.0,11.25,15.0 Seed:8 OK:-1.0,4.0,5.0,0.0,3.75,6.0
cmi.interactions_0.student_response 	52,54,7,5,2,21 5,9,8,2,3,2 9,9,9,9,9
cmi.objectives_0.id 	Puntuacion: 1 0 0

Descargar datos de tabla como Archivo de texto con valores separados por comas Descargar

Fig. 3.3 Forma en la que el profesor recibe los datos de la resolución de un ejercicio por parte de un alumno

Si el alumno realiza otro ejercicio con la misma sesión los datos se almacenarán a continuación de los del primer ejercicio con la misma estructura. Y así para todos los ejercicios realizados durante una misma sesión.

Al finalizar la sesión se enviarán los resultados almacenados de todos los ejercicios realizados por el alumno.

El envío de los datos se realiza desde el módulo ComunicaciónConLMS, pero el almacenamiento se realiza en el módulo GestorDeSesión.

La solución que damos para el ejercicio continuo es exactamente la misma que se ha explicado en este apartado como solución para el ejercicio discreto, por ello, aunque hemos realizado cambios en el módulo GestorDeSesion de ejercicio continuo, no se van a volver a explicar.

3.3.2 Ejercicio continuo.

Al igual que en el ejercicio discreto, nos dedicamos a corregir errores y depurar el funcionamiento del ejercicio.

3.3.2.1 Generación del enunciado

La idea es la misma que en el ejercicio discreto, ya que a partir de una semilla obtendremos los diversos parámetros que componen las dos señales con las que trabajaremos en el ejercicio. El enunciado se compone de dos señales LTI de la forma:

$$x[t] = A_1 t^N (u[t - t_{11}] - u[t - t_{12}]); \quad (14)$$

$$h[t] = A_2 t^{(1-N)} (u[t - t_{21}] - u[t - t_{22}]); \quad (15)$$

Para calcular los parámetros del ejercicio utilizamos la semilla aplicando las mismas fórmulas que en el apartado de discreto. (Ver fórmulas de la 3 a la 10).

Los valores de los parámetros t_{11} , t_{12} , t_{21} , t_{22} , A_1 , A_2 y N de estas funciones a partir de la semilla se obtienen de la siguiente forma.

Algoritmo de la obtención del enunciado:

Para obtener $x[t]$

Escribimos “ $x[t]=(2+(Mm3/2))$ ”

Si($M2$ es igual a 1)

Escribo “ t “

Si($M2$ es distinto de 0)

Escribo “ t^{M2} “

Escribo “ [$u(t$ “

Si((($2-(M7/2)$) mayor que 0)

Escribimos “+”

Si((($2-(M7/2)$) distinto de 0)

Escribimos “ $2-(M7/2)$ ”

Escribimos “ $)]-u[n$ ”

Si(la semilla no es múltiplo de los dividendos)

Si((($-M7/2)-Mm4$) es distinto de 0)

Escribimos “ $(-M7/2)-Mm4$ ”

Obtenemos un $x[t]$ de la forma:

$$x[t] = \frac{2+Mm3}{2} t^{M2} \left(u \left[t + 2 - \frac{M7}{2} \right] - u \left[t - \frac{M7}{2} - Mm4 \right] \right); \quad (16)$$

Siendo esta una forma particular de la ecuación (14).

Para obtener $h[t]$

Escribimos “ $h[t]=((2+Mm^4)/2)$ ”

Si (M^2) es igual a 1)

Escribo “ t “

Si (M^2) es distinto de 0)

Escribo “ $t^{(1-M^2)}$ “

Si (M^2) es igual a 1)

Escribimos “ t ”

Sino

Escribimos “ t^{M^2} ”

Escribimos “ $t[n]$ ”

Si $((1-(M^5/2))$ es mayor que 0)

Escribimos “ + “

Si $((1-(M^5/2))$ es distinto de 0)

Escribimos “ $1-(M^5/2)$ ”

Escribimos “ $] - u[t$ “

Escribimos “ $-(M^5/2)-M^3]$ “

Obtenemos un $h[t]$ de la forma:

$$h[t] = \left(\frac{2+Mm^4}{2}\right)t^{(1-M^2)} \left(u \left[t + 1 - \frac{M^5}{2}\right] - u \left[t - \frac{M^5}{2} - M^3\right]\right); \quad (17)$$

Siendo esta una forma particular de la ecuación (15).

3.3.2.2 Obtención de la convolución en tiempo continuo.

Partiendo de las ecuaciones obtenidas anteriormente podemos deducir que tanto $x[t]$ como $h[t]$ pueden tomar tanto en forma rectangular como de rampa tal como se observa en la figura 3.4 para el caso del tiempo continuo, pero nunca coinciden en la forma.

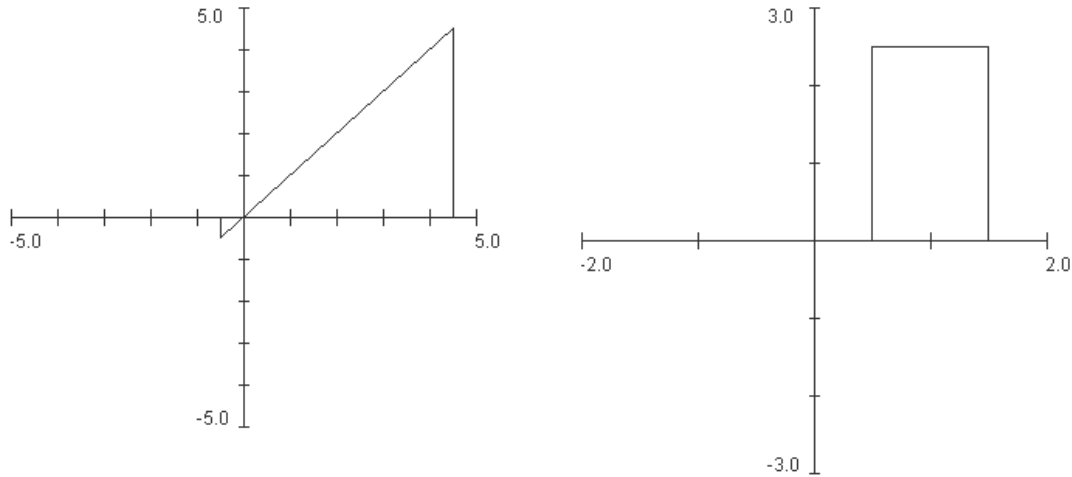


Fig. 3.4 Tipo de funciones en tiempo continuo.

A partir de este momento, para el cálculo de la convolución de las dos señales LTI nos referiremos a $x[t]$ como $X1$ y a $h[t]$ como $X2$.

$$x[t] \rightarrow X1 = A_1 t^N (u[t - t_{11}] - u[t - t_{12}]); \quad (18.a)$$

$$h[t] \rightarrow X2 = A_2 t^{(1-N)} (u[t - t_{21}] - u[t - t_{22}]); \quad (19.a)$$

Donde los parámetros se obtienen comparando las ecuaciones (16) y (17) con (14) y (15) respectivamente. Utilizando el vector “p” de la ecuación (11) el resultado es:

$$A_1 = \frac{2 + p[5]}{2}; \quad (20)$$

$$A_2 = \frac{2 + p[6]}{2}; \quad (21)$$

$$t_{11} = -2 + \frac{p[3]}{2}; \quad (22)$$

$$t_{12} = \frac{p[3]}{2} + p[6]; \quad (23)$$

$$t_{21} = -1 + \frac{p[2]}{2}; \quad (24)$$

$$t_{22} = \frac{p[2]}{2} + p[1]; \quad (25)$$

Algoritmo:

Este se puede dividir en dos grandes bloques diferenciando la forma de las señales.

Bloque 1.- Cuando $x(t)$ es una rampa y $h(t)$ es un rectángulo

$$x(t) \rightarrow A_1 t [u(t - t_{11}) - u(t - t_{12})]; \quad (18. b)$$

$$h(t) \rightarrow A_2 [u(t - t_{21}) - u(t - t_{22})]; \quad (19. b)$$

Comprobamos que estamos en el rango donde coinciden las señales

Si ($h(t)$ es más estrecho que $x(t)$)

Cuando $h(t)$ esta “entrando” en $x(t)$

$$y(t) = A_1 A_2 \left[\frac{(t - t_{21})^2 - t_{11}^2}{2} \right];$$

Cuando $h(t)$ esta “dentro” de $x(t)$

$$y(t) = A_1 A_2 \left[t(t_{22} - t_{21}) - \frac{t_{22}^2}{2} + \frac{t_{21}^2}{2} \right];$$

Cuando $h(t)$ está “saliendo” de $x(t)$

$$y(t) = A_1 A_2 \left[\frac{t_{12}^2 - (t - t_{22})^2}{2} \right];$$

Si ($h(t)$ es más ancho que $x(t)$)

Cuando $h(t)$ esta “entrando” en $x(t)$

$$y(t) = A_1 A_2 \left[\frac{(t - t_{21})^2 - t_{11}^2}{2} \right];$$

Cuando $h(t)$ esta “dentro” de $x(t)$

$$y(t) = \frac{A_1 A_2}{2} [t_{12}^2 - t_{11}^2];$$

Cuando $h(t)$ está “saliendo” de $x(t)$

$$y(t) = A_1 A_2 \left[\frac{t_{12}^2 - (t - t_{22})^2}{2} \right];$$

Bloque 2.- Cuando $x(t)$ es un pulso rectángulo y $h(t)$ una rampa.

$$x(t) \rightarrow A_1[u(t - t_{11}) - u(t - t_{12})]; \quad (18. c)$$

$$h(t) \rightarrow A_2 t[u(t - t_{21}) - u(t - t_{22})]; \quad (19. c)$$

Comprobamos que estamos en el rango donde coinciden las señales

Si $h(t)$ es más estrecho que $x(t)$

Cuando $h(t)$ esta “entrando” en $x(t)$

$$y(t) = \frac{A_1 A_2}{2} [(t - t_{11})^2 - t_{21}^2];$$

Cuando $h(t)$ esta “dentro” de $x(t)$

$$y(t) = \frac{A_1 A_2}{2} [t_{22}^2 - t_{21}^2];$$

Cuando $h(t)$ está “saliendo” de $x(t)$

$$y(t) = \frac{A_1 A_2}{2} [t_{22}^2 - (t - t_{12})^2];$$

Si $h(t)$ es más ancho que $x(t)$

Cuando $h(t)$ esta “entrando” en $x(t)$

$$y(t) = \frac{A_1 A_2}{2} [(t - t_{11})^2 - t_{21}^2];$$

Cuando $h(t)$ esta “dentro” de $x(t)$

$$y(t) = \frac{A_1 A_2}{2} [(t - t_{21})^2 - (t - t_{12})^2];$$

Cuando $h(t)$ está “saliendo” de $x(t)$

$$y(t) = \frac{A_1 A_2}{2} [t_{22}^2 - (t - t_{12})^2];$$

3.3.4 Ejercicio gráficas

Este ejercicio es el mayor reto al que nos enfrentábamos en este PFC. Nuestro tutor quería que lográsemos una dinamización del ejercicio, pudiendo tener varias presentaciones de cara al usuario.

Esto se tenía que reflejar en la forma de la presentación del enunciado, ya que las funciones con las que el usuario tiene que trabajar, se puedan presentar de forma gráfica o de ecuación.

Como ya comentamos en la introducción el trabajar con el lenguaje Java y sus clases, nos permite poder trabajar sobre ejercicios ya existentes añadiendo, quitando o modificando módulos, para obtener un nuevo ejercicio de una forma sencilla y rápida.

Para poder encarar este reto, hemos realizado el siguiente trabajo:

- Módulo Enun_Manager (Fig. 3.2): Realizamos modificaciones en el mismo para que nos devuelva los parámetros de la ecuación que necesitamos para poder ser representada de forma gráfica.
- Applet: Modificamos el código añadiendo dos nuevos paneles para representar cada una de las funciones, las cuales se mostrarán de forma textual o grafica dependiendo de la semilla introducida.
- Módulo Figuras: Se ha generado este módulo para poder hacer las representaciones graficas de las funciones.

3.3.4.1 Modificaciones Módulo Enun_Manager

En este módulo se han realizado dos grandes cambios:

Para poder representar las funciones de diferente forma, se ha dividido la forma de redactar el enunciado en dos partes para poder separar la función x de la función h . Estas funciones son devueltas mediante las funciones “escribirEnunciadox” y “escribirEnunciadoh”.

Se añaden dos nuevas funciones `getXt` y `getHt`, las cuales devuelven los parámetros ($t1$, $t2$, altura, triangular, tipo). Estos parámetros son obtenidos mientras la función “obtenerenunciado” redacta el enunciado. Gracias a estos parámetros nos será posible determinar:

- Tiempo inicial
- Tiempo final
- Altura o pendiente: Dependiendo de si la función es triangular o rectangular este parámetro determinará la altura o la pendiente de la misma.

- Triangular o rectangular
- El tipo de función (discreto o continuo).

3.3.4.2 Modificaciones en el Applet

Los cambios realizados en este apartado afectan a la forma de expresar el enunciado. Esta tarea se engloba dentro de la función `NuevoActionPerformed`; esta función se encarga de, a partir de la semilla introducida comunicarse con los diferentes módulos que componen el programa para obtener el enunciado completo que el alumno debe resolver.

Como el objetivo principal de este programa, era poder añadir la posibilidad de mostrar de forma gráfica las funciones del enunciado, hemos tenido que dividir el enunciado en tres bloques principales:

Encabezamiento.- Pequeña introducción de lo que pretende el ejercicio.

Funciones.- Presenta las funciones en las que se basa el ejercicio según la semilla introducida.

Apartados del ejercicio.- Enumeración de los problemas a resolver.

Tanto los bloques “Encabezamiento” como “Apartados del ejercicio” no se modifican.

3.3.4.2.1 Modificaciones en el bloque Funciones.

El pseudocódigo que representa los cambios realizados en este bloque es el siguiente:

Dividimos la presentación de las funciones en dos paneles más pequeños.

Obtenemos la función x

Obtenemos la función h

Generamos la gráfica x

Generamos la gráfica h

Generamos el enunciado en texto de x

Generamos el enunciado en texto de h

La apariencia del enunciado depende de la semilla introducida. Podemos mostrar la función tanto en texto como en figura, la forma final dependerá de la semilla introducida (nos quedamos con el resto de la división entre 4), de esta manera conseguiremos una aleatoriedad en la representación del enunciado.

3.3.4.3 Módulo figuras

Este módulo es totalmente nuevo, con él podemos obtener la gráfica de cualquier función de tipo escalón o rampa tanto en discreto como en continuo. Representa perfectamente la filosofía Java ya que puede ser usado en cualquier otro programa, teniendo en cuenta los parámetros que necesita. Estos parámetros son:

Momento inicial.- El momento en el cual comienza la representación de la función.

Momento final.- El momento en el cual termina la representación de la función.

Altura/Pendiente.- Dependiendo del tipo de función este parámetro recogerá la altura si la función a representar es un escalón, o la pendiente si la misma fuese triangular.

Triangular/escalón.- Parámetro binario que nos indica la naturaleza de la función.

Tipo.- Parámetro binario que nos muestra si el sistema a estudiar es discreto o continuo.

Además de los parámetros que acabamos de enunciar, a la clase Figuras, habrá que pasarle el tamaño del panel donde representaremos la gráfica, y el punto inicial del mismo.

Dentro de este módulo usamos tres funciones dentro de la clase Graphics para poder pintar las gráficas: Estas funciones son:

- drawString: Dibuja el texto del String en las coordenadas indicadas

Parámetros:	String cadena	->	Texto a dibujar
	int x	->	Coordenada X
	int y	->	Coordenada Y

- drawChars: Dibuja un array de caracteres en las coordenadas indicadas

Parámetros:	char[] dato	->	Array de caracteres a dibujar
	int offset	->	Posición inicial del array
	int length	->	Longitud de caracteres a
	pintar		
	int x	->	Coordenada X
	int y	->	Coordenada Y

- drawPolyLine: Dibuja una serie de líneas conectadas en base a un array de coordenadas x y otro de coordenadas y.

Parámetros:	int[] X	->	Array coordenadas X
	int[] Y	->	Array coordenadas Y
	int Puntos	->	Número de puntos a pintar

Podemos observar que todas estas funciones tienen como parámetro de entrada unas coordenadas, las cuales están referenciadas sobre la esquina superior izquierda del tapiz de trabajo.

Pseudocódigo del módulo figuras:

Obtenemos los parámetros.

Calculamos la resolución de ambos ejes cartesianos.

$$\text{Eje } y = \frac{150}{\text{altura} + 1};$$

La resolución del eje x se calcula atendiendo al momento inicial o final, dependiendo de cuál sea el de mayor valor absoluto.

$$\text{Eje } x = \frac{150}{t + 1};$$

Calculamos las divisiones de ambos ejes.

Calculamos el punto de referencia para poder dibujar, el origen de coordenadas.

Si es triangular.

Respecto a los datos obtenidos anteriormente, si la señal es triangular, varían la resolución, el número de divisiones y el origen de coordenadas sólo respecto del eje y.

En este caso, recibimos como variable de entrada la pendiente, lo que provoca que la altura de la figura sea variable.

Una vez tenemos los datos para enmarcar las figuras.

Si es continuo

{

Si es triangular

 Calculo la altura en el momento inicial y final de la señal

 Pinto la función triangular

Si no

 Pinto la función rectangular

}

En la parte de discreto la forma de dibujar es más compleja, ya que el número de líneas es mayor. Como ya hemos comentado anteriormente, usamos la función drawPolyLine que básicamente genera líneas entre dos coordenadas dadas.

La forma en que resolvemos el problema, es haciendo una sola llamada a la función drawPolyLine, por lo tanto, generamos dos arrays donde almacenaremos todas las posibles coordenadas a representar, y más tarde le pasaremos estos arrays con toda la información de la señal, para pintar la función directamente.

Para pintar cada una de las muestras necesitamos tres coordenadas (Fig. 3.5).

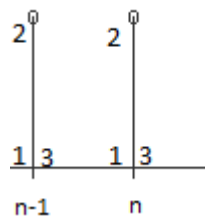


Fig. 3.5 Secuencia de coordenadas. Donde 1,2 y3 son las distintas coordenadas que necesitamos almacenar.

Pintamos siguiendo la siguiente secuencia:

- 1.- Unimos la coordenada 3 de la muestra “n-1” con la coordenada 1 de “n”.
- 2.- Unimos la coordenada 1 con la 2 de la muestra “n”.
- 3.- Unimos la coordenada 2 con la 3.

Los arrays que generamos tienen 33 posiciones. Esto se debe a que nuestras funciones están comprendidas entre un rango de “-5 a 5”, por lo que hay un máximo de 11 muestras con tres coordenadas por muestra.

Si es discreto

{

 Si es rectangular

 {

 Almacenamos las coordenadas para cada muestra.

 Pintamos la función.

 }

Si no

{

Calculamos la altura de las muestras en los extremos.

Calculamos la variación de altura entre muestras, lo que dependerá de la pendiente que tenemos como parámetro de entrada.

Almacenamos las coordenadas de cada una de las muestras.

Pintamos la función.

}

}

Pintamos los guiones de cada uno de los ejes.

FIN

4. PRUEBAS Y RESULTADOS

4.1 Ejercicio discreto

El problema a resolver, tanto en la parte discreta como en la continua, era la compatibilidad con introducir como semilla algún múltiplo de los números usados en el algoritmo de la resolución del enunciado. Con las explicaciones aportadas en el capítulo tres (ver pag 36) el resultado ahora es el siguiente para el ejercicio discreto.

Problema: [Convolucion analitica](#)

Dado un sistema LTI de respuesta al impulso $h[n]$, obtener la salida $y[n]$ para una entrada $x[n]$.

$$x[n] = (u[n+3] - u[n+1])$$
$$h[n] = (u[n+2] - u[n-1])$$

a) Determinar el primer instante en el que $y[n]$ es distinto de cero.

b) Determinar el ultimo instante en el que $y[n]$ es no nulo.

c) Determinar la duracion de $y[n]$.

d) Dar los valores de $y[n]$ para:

n = -2	<input type="text"/>
n = -1	<input type="text"/>
n = 0	<input type="text"/>

Introduce un Nº

Fig. 4.1 Ejercicio discreto corregido el error números múltiplos.

El número 30030 cumple con los requisitos nombrados anteriormente, y se aprecia que el enunciado es correcto.

4.2 Ejercicio continuo.

Para el ejercicio continuo, teníamos dos focos problemáticos; el error de la semilla múltiplo de los números del algoritmo, y hacer funcionar la parte de la convolución de manera correcta.

La resolución al primero de los problemas que resuelta como se indicó en el capítulo 3 (ver pag 40).

Problema: Convolucion analitica

Dado un sistema LTI de respuesta al impulso $h(t)$, obtener la salida $y(t)$ para una entrada $x(t)$.

$$x(t) = [u(t+2.0) - u(t)]$$
$$ecoh(t) = t[u(t+1.0) - u(t)]$$

a) Determinar el primer instante en el que $y(t)$ es distinto de cero.

b) Determinar el ultimo instante en el que $y(t)$ es no nulo.

c) Determinar la duracion de $y(t)$.

d) Dar los valores de $y(t)$ para:

$t = -3.0$

$t = -1.5$

$t = -1.0$

Calificación:

Introduce un Nº

Fig. 4.2 Ejercicio continuo corregido el error números múltiples.

Para comprobar el buen funcionamiento del cálculo de la convolución haremos dos pruebas:

Semilla 30030

Problema: Convolucion analitica

Dado un sistema LTI de respuesta al impulso $h(t)$, obtener la salida $y(t)$ para una entrada $x(t)$.

$$x(t) = [u(t+2.0) - u(t)]$$

$$h(t) = t[u(t+1.0) - u(t)]$$

a) Determinar el primer instante en el que $y(t)$ es distinto de cero. CORRECTO

b) Determinar el ultimo instante en el que $y(t)$ es no nulo. CORRECTO

c) Determinar la duracion de $y(t)$. CORRECTO

d) Dar los valores de $y(t)$ para:

$t = -3.0$	<input type="text" value="0"/>	CORRECTO
$t = -1.5$	<input type="text" value="-0.5"/>	CORRECTO
$t = -1.0$	<input type="text" value="-0.5"/>	CORRECTO

Calificación: 6

Introduce un Nº

Fig. 4.3 Ejercicio continuo comprobación de la convolución para la semilla 30030.

Valores correctos para la convolución de la semilla 30030:

$$t = -3 \rightarrow y(t) = 0.$$

$$t = -1.5 \rightarrow y(t) = -0.5.$$

$$t = -1.0 \rightarrow y(t) = -0.5.$$

Con lo que se comprueba que la corrección es perfecta.

Semilla 58

Problema: Convolucion analitica

Dado un sistema LTI de respuesta al impulso $h(t)$, obtener la salida $y(t)$ para una entrada $x(t)$.

$$x(t) = [u(t+1.0) - u(t-3.0)]$$

$$h(t) = 2.0t[u(t-0.5) - u(t-2.5)]$$

a) Determinar el primer instante en el que $y(t)$ es distinto de cero. INCORRECTO

b) Determinar el ultimo instante en el que $y(t)$ es no nulo. CORRECTO

c) Determinar la duracion de $y(t)$. CORRECTO

d) Dar los valores de $y(t)$ para:

$t = 1.5$	<input type="text" value="6"/>	CORRECTO
$t = 3.0$	<input type="text" value="6"/>	CORRECTO
$t = 3.5$	<input type="text" value="0"/>	INCORRECTO

Calificación: 4

Introduce un Nº

Fig. 4.4 Ejercicio continuo comprobación de la convolución para la semilla 58.

Valores correctos de la convolución para la semilla 58:

$$t = 1.5 \rightarrow y(t) = 6.$$

$$t = 3.0 \rightarrow y(t) = 6.$$

$$t = 3.5 \rightarrow y(t) = 6.$$

Por lo que se comprueba que la corrección es perfecta.

4.3 Ejercicio gráficas

Lo más importante y que claramente diferencia este ejercicio con los anteriores es el dinamismo del enunciado, es decir, ahora podemos mostrar el enunciado no solo como enunciados escritos, sino que podemos mostrarlos como la gráfica que los representa.

El ejercicio de las gráficas podríamos dividirlo igualmente en discreto y continuo, para mostrar mejor el correcto funcionamiento del mismo.

4.3.1 Ejercicio discreto con gráficas.

Tenemos tres posibles formatos de enunciados:

Clásico: Ambas funciones se muestran de forma analítica

Dado un sistema LTI de respuesta al impulso $h[n]$, obtener la salida $y[n]$ para una entrada $x[n]$.
 Nota: Fuera del intervalo representado las señales son nulas.

$x[n] = 2(u[n-1] - u[n-3])$ $h[n] = (u[n-2] - u[n-6])$

a) Determinar el primer instante en el que $y[n]$ es distinto de cero.

b) Determinar el último instante en el que $y[n]$ es no nulo.

c) Determinar la duración de $y[n]$.

d) Dar los valores de $y[n]$ para:

$n = 6$

$n = 7$

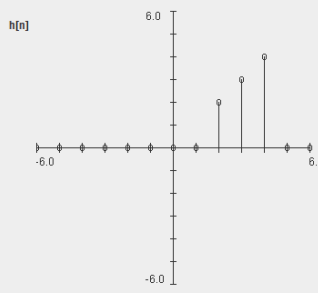
$n = 8$

Introduce un N°

Fig. 4.5 Enunciado con ambas funciones de forma analítica.

Gráfica y función: Se muestra una en forma analítica y la otra en forma gráfica.

Dado un sistema LTI de respuesta al impulso $h[n]$, obtener la salida $y[n]$ para una entrada $x[n]$.
 Nota: Fuera del intervalo representado las señales son nulas.



$x[n] = (u[n+1] - u[n-2])$

a) Determinar el primer instante en el que $y[n]$ es distinto de cero.

b) Determinar el último instante en el que $y[n]$ es no nulo.

c) Determinar la duración de $y[n]$.

d) Dar los valores de $y[n]$ para:

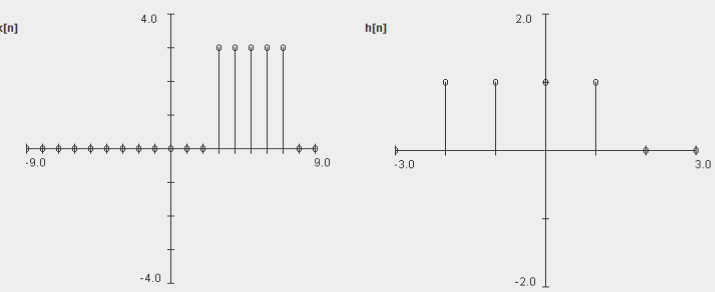
	$n = 4$	<input type="text"/>
	$n = 5$	<input type="text"/>
	$n = 6$	<input type="text"/>

Introduce un Nº

Fig. 4.6 Enunciado con una función en forma analítica y la otra en forma gráfica.

Ambas gráficas: Las dos funciones se muestran de forma gráfica.

Dado un sistema LTI de respuesta al impulso $h[n]$, obtener la salida $y[n]$ para una entrada $x[n]$.
 Nota: Fuera del intervalo representado las señales son nulas.



a) Determinar el primer instante en el que $y[n]$ es distinto de cero.

b) Determinar el último instante en el que $y[n]$ es no nulo.

c) Determinar la duración de $y[n]$.

d) Dar los valores de $y[n]$ para:

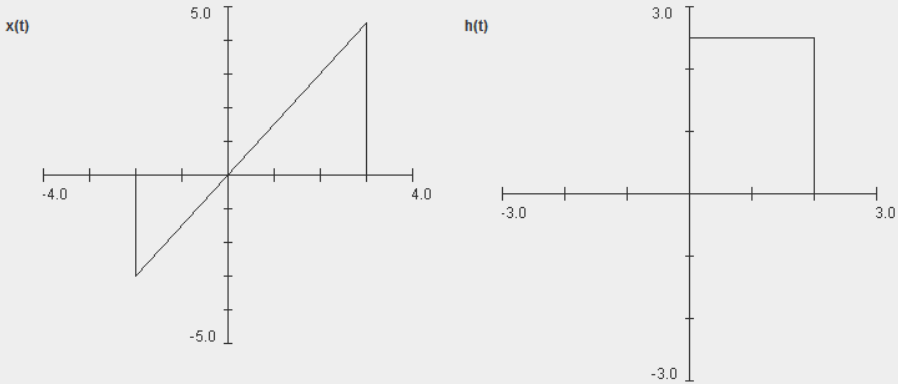
	$n = 4$	<input type="text"/>
	$n = 5$	<input type="text"/>
	$n = 6$	<input type="text"/>

Introduce un Nº

Fig. 4.7 Enunciado con ambas funciones en forma gráfica.

Ambas gráficas: Las dos funciones se muestran de forma gráfica.

Dado un sistema LTI de respuesta al impulso $h(t)$, obtener la salida $y(t)$ para una entrada $x(t)$.
Nota: Fuera del intervalo representado las señales son nulas.



a) Determinar el primer instante en el que $y(t)$ es distinto de cero.

b) Determinar el último instante en el que $y(t)$ es no nulo.

c) Determinar la duración de $y(t)$.

d) Dar los valores de $y(t)$ para:

$t = 1.0$	<input type="text"/>
$t = 2.5$	<input type="text"/>
Calificación: $t = 3.0$	<input type="text"/>

Introduce un Nº

Fig. 4.10 Enunciado con ambas funciones en forma gráfica

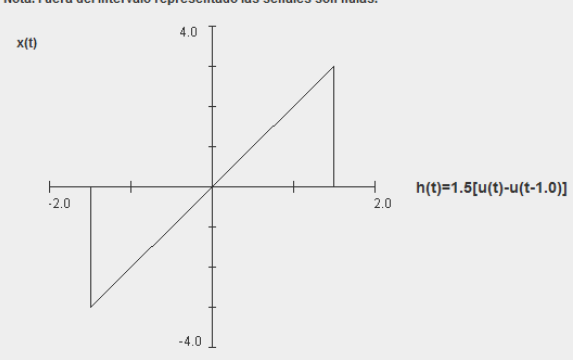
4.4 Almacenamiento de datos.

Una de las partes más importantes del PFC es poder capturar y almacenar todos los datos posibles de los alumnos al resolver los ejercicios, para que el profesor pueda disponer de ellos para realizar un seguimiento de los alumnos.

Vamos a mostrar cómo se hace en el supuesto de que un alumno haga tres ejercicios introduciendo sus tres respectivas semillas.

Ejercicio 1. Semilla 57

Dado un sistema LTI de respuesta al impulso $h(t)$, obtener la salida $y(t)$ para una entrada $x(t)$.
 Nota: Fuera del intervalo representado las señales son nulas.



a) Determinar el primer instante en el que $y(t)$ es distinto de cero. CORRECTO

b) Determinar el último instante en el que $y(t)$ es no nulo. CORRECTO

c) Determinar la duración de $y(t)$. CORRECTO

d) Dar los valores de $y(t)$ para:

$t = -0.5$	<input type="text" value="-3"/>	CORRECTO
$t = 1.0$	<input type="text" value="1"/>	INCORRECTO
$t = 1.5$	<input type="text" value="3"/>	CORRECTO

Calificación: 5

Fig. 4.11 Ejemplo con semilla 57.

Respuestas correctas:

a) $y(t) = -1.5$

b) $y(t) = 4.5$

c) $y(t) = 6$

d) $t = -0.5 \rightarrow y(t) = -3$

$t = 1 \rightarrow y(t) = 1.5$

$t = 1.5 \rightarrow y(t) = 3$

Ejercicio 2. Semilla 12

Dado un sistema LTI de respuesta al impulso $h(t)$, obtener la salida $y(t)$ para una entrada $x(t)$.
Nota: Fuera del intervalo representado las señales son nulas.

$x(t)=[u(t-0.5)-u(t-2.5)]$ $h(t)=t[u(t)-u(t-1.0)]$

a) Determinar el primer instante en el que $y(t)$ es distinto de cero. CORRECTO

b) Determinar el ultimo instante en el que $y(t)$ es no nulo. CORRECTO

c) Determinar la duracion de $y(t)$. CORRECTO

d) Dar los valores de $y(t)$ para:

$t = 0.5$	<input type="text" value="0"/>	CORRECTO
$t = 2.0$	<input type="text" value="0.5"/>	CORRECTO
$t = 2.5$	<input type="text" value="0.5"/>	CORRECTO

Calificación: 6

Introduce un Nº

Fig. 4.12 Ejemplo con semilla 12.

Respuestas correctas:

a) $y(t) = 0.5$

b) $y(t) = 3.5$

c) $y(t) = 3$

d) $t = 0.5 \rightarrow y(t) = 0$

$t = 2 \rightarrow y(t) = 0.5$

$t = 2.5 \rightarrow y(t) = 0.5$

Ejercicio 3. Semilla 32

Dado un sistema LTI de respuesta al impulso $h(t)$, obtener la salida $y(t)$ para una entrada $x(t)$.
 Nota: Fuera del intervalo representado las señales son nulas.

a) Determinar el primer instante en el que $y(t)$ es distinto de cero. **CORRECTO**

b) Determinar el último instante en el que $y(t)$ es no nulo. **INCORRECTO**

c) Determinar la duración de $y(t)$. **INCORRECTO**

d) Dar los valores de $y(t)$ para:

$t = 2.0$	<input type="text" value="5"/>	INCORRECTO
$t = 3.5$	<input type="text" value="10"/>	INCORRECTO
$t = 4.0$	<input type="text" value="10"/>	INCORRECTO

Calificación: 1

Fig. 4.13 Ejemplo con semilla32

Respuestas correctas:

a) $y(t) = 0$

b) $y(t) = 5$

c) $y(t) = 5$

d) $t = 2 \rightarrow y(t) = 4$

$t = 3.5 \rightarrow y(t) = 9$

$t = 4 \rightarrow y(t) = 9$

Una vez que el alumno a contestado los apartados de los ejercicios, los datos quedan almacenados en Moodle de la siguiente manera:





Elemento	Valor
cmi.core.lesson_status	completed
cmi.core.score.raw 	5
cmi.interactions_0.id 	Seed:57 OK:- 1.5,4.5,6.0,-3.0,1.5,3.0 Seed:12 OK:0.5,3.5,3.0,0.0,0.5,0.5 Seed:32 OK:0.0,5.0,5.0,4.0,9.0,9.0
cmi.interactions_0.student_response 	-1.5,4.5,6,-3,1,3 0.5,3.5,3,0,0.5,0.5 0,4,4,5,10,10
cmi.objectives_0.id 	Puntuacion: 5 6 1

Fig. 4.14 Datos alumno.

cmi.core.score.raw: Almacena la puntuación del primero de los ejercicios con la primera semilla.

cmi.interactions_0.id: Almacena la semilla y las respuestas correctas para esta semilla con el siguiente formato:

Seed: "semilla" | OK: "Respuesta correcta apdo a)","respuesta correcta apdo b)","respuesta correcta apdo c)", "respuesta correcta apdo d)1", "respuesta correcta apdo d)2", "respuesta correcta apdo d)3" | Seed: "semilla".....

cmi.interactions_0.student_response: Almacena la respuesta del alumnos a cada uno de los apartados con el siguiente formato:

"Respuesta alumno apdo a)","respuesta alumno apdo b)", "respuesta alumno apdo c)", "respuesta alumno apdo d)1", "respuesta alumno apdo d)2", "respuesta alumno apdo d)3" | "Respuesta alumno apdo a)","respuesta alumno apdo b)",.....

cmi.objectives_0.id: Almacena las puntuaciones de todos los ejercicios realizados por el alumno, con el siguiente formato:

Puntuación: " Puntuación primer ejercicio" | "Puntuación segundo ejercicio"|

Moodle también le da la posibilidad al profesor de descargarse la tabla en varios formatos a elegir.

Para el ejemplo que nos ocupa, de la tabla podemos obtener los valores del ejercicio hecho por el alumno de manera sencilla:

	Respuestas _OK		Respuestas Alumno	Calificación
Ejercicio 1 Semilla 57	-1.5		-1.5	5
	4.5		4.5	
	6		6	
	Apdo. d	-3	-3	
		1.5	1	
3		3		
Ejercicio 2 Semilla 12	0.5		0.5	6
	3.5		3.5	
	3		3	
	Apdo. d	0	0	
		0.5	0.5	
0.5		0.5		
Ejercicio 3 Semilla 32	0		0	1
	5		4	
	5		4	
	Apdo. d	4	5	
		9	10	
9		10		

5. ESTUDIO DEL IMPACTO DE LA APLICACIÓN EN ALUMNOS

Una vez terminado el desarrollo de las aplicaciones y habiendo hecho las comprobaciones oportunas para verificar un correcto funcionamiento, nos disponemos a realizar unas pruebas con alumnos.

Sobre una muestra de 6 alumnos, realizaremos un pequeño estudio del posible impacto de la realización del ejercicio sobre su nota en un examen. Los alumnos han tenido disponible el ejercicio en sus dos variantes, la versión discreta y continua.

5.1 Ejercicio discreto

Para la realización de estas pruebas disponemos de cinco alumnos, a los que nos referiremos como “alumno1”, “alumno2”, y “alumno3”, “alumno4”, “alumno5”.

A todos los alumnos se les dan las mismas pautas para la realización del ejercicio. Pueden usar semillas, a su elección y que pueden hacer todos los intentos que quieran.

El alumno1 elige la semilla “1”, y realiza 16 intentos hasta alcanzar la máxima puntuación del ejercicio.

El alumno2 elige la semilla “19”, y realiza 9 intentos hasta alcanzar la máxima puntuación del ejercicio.

El alumno3 elige la semilla “3”, y realiza 7 intentos hasta alcanzar la máxima puntuación del ejercicio.

El alumno4 elige la semilla “7”, y realiza 3 intentos hasta alcanzar la máxima puntuación del ejercicio.

El alumno 5 elige dos semillas, los números “1” y “2”, y sólo necesita un intento con cada una para obtener la puntuación máxima.

En la siguiente gráfica se observa con claridad la evolución.

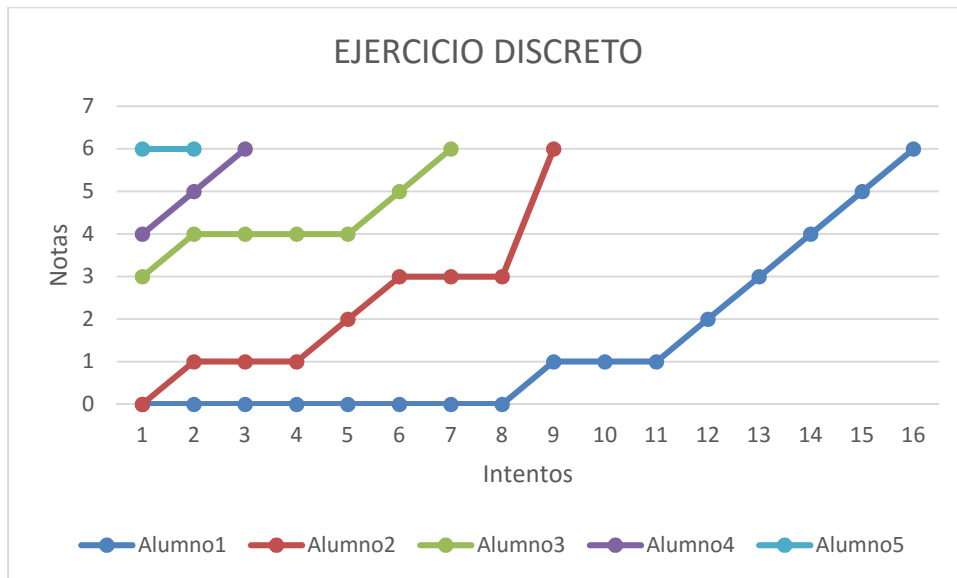


Fig. 5.1 Evolución de los alumnos en el ejercicio discreto.

También, para el ejercicio discreto, tanto el alumno2 como el alumno4, realizaron varios intentos con diferentes semillas. Hemos realizado unas graficas que muestran las notas obtenidas en estos intentos sin importar la semilla usada.

El alumno2 realizó un total de 12 intentos con el siguiente resultado:

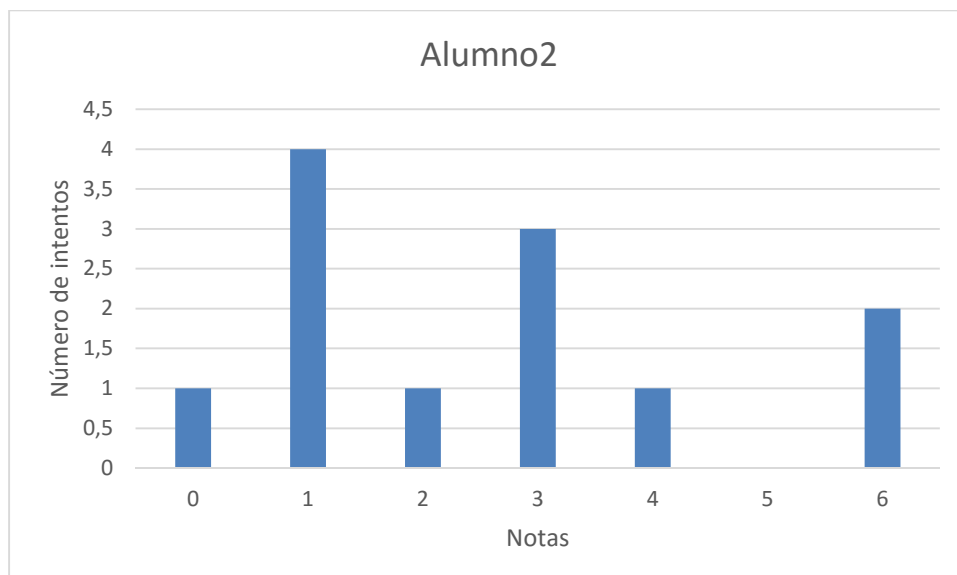


Fig. 5.2 Resultados del alumno2.

El alumno4 realizó un total de 19 intentos con el siguiente resultado:

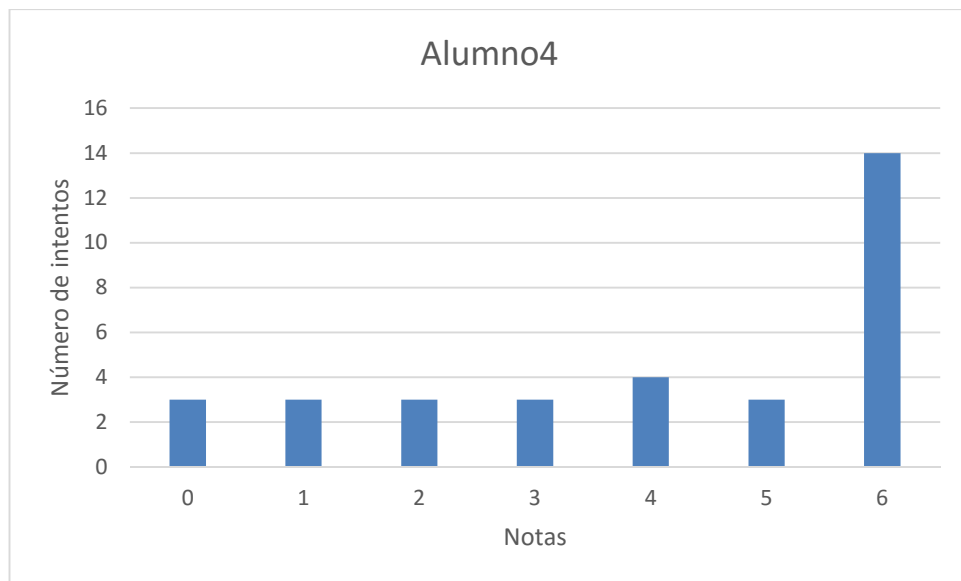


Fig. 5.3 Resultados del alumno4.

5.2 Ejercicio continuo

Para la realización de estas pruebas disponemos de cinco alumnos, a los que nos referiremos como “alumno1”, “alumno2”, “alumno4”, “alumno5”, y “alumno6”.

A todos los alumnos se les dan las mismas pautas para la realización del ejercicio. Pueden usar semillas a su elección y que pueden hacer todos los intentos que quieran.

El alumno1 elige la semilla “4”, y realiza 6 intentos hasta alcanzar una puntuación máxima del ejercicio

El alumno2 elige la semilla “5”, y realiza 4 intentos hasta alcanzar una puntuación máxima de 4.

El alumno4 elige la semilla “0”, y realiza 5 intentos hasta alcanzar una puntuación máxima de 4.

El alumno5 elige la semilla “3”, y realiza 5 intentos hasta alcanzar una puntuación máxima de 3.

El alumno6 elige la semilla “3”, y realiza 17 intentos hasta alcanzar una puntuación máxima de 5.

En la siguiente gráfica se observa claramente la evolución.

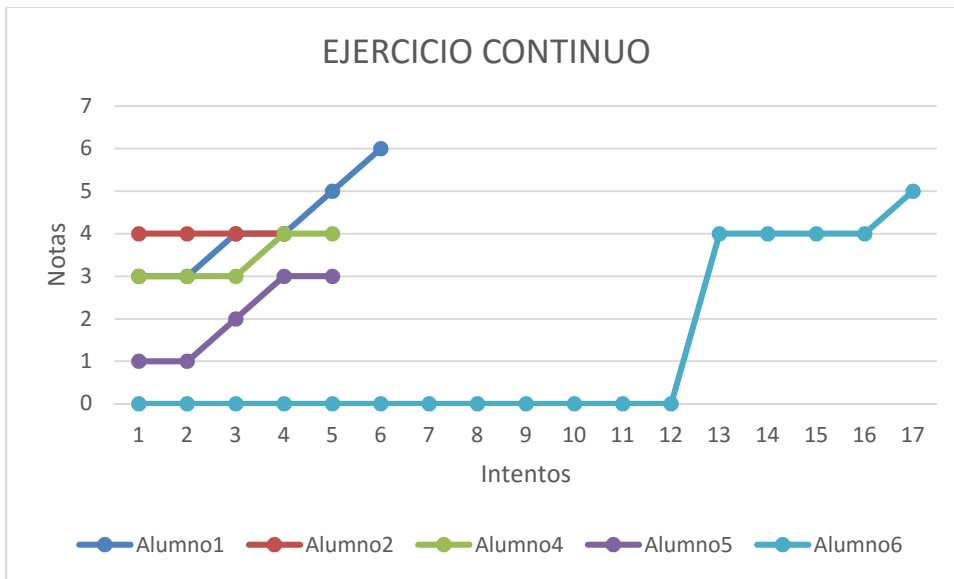


Fig. 5.4 Evolución de los alumnos en el ejercicio continuo.

Tal como se sucedió en el ejercicio discreto, para el continuo tanto el alumno1 como el alumno6 realizaron diversos intentos sin importar la semilla que usaban. Las gráficas que representan las notas obtenidas por ambos son las siguientes:

El alumno1 realizó un total de 29 intentos con el siguiente resultado:

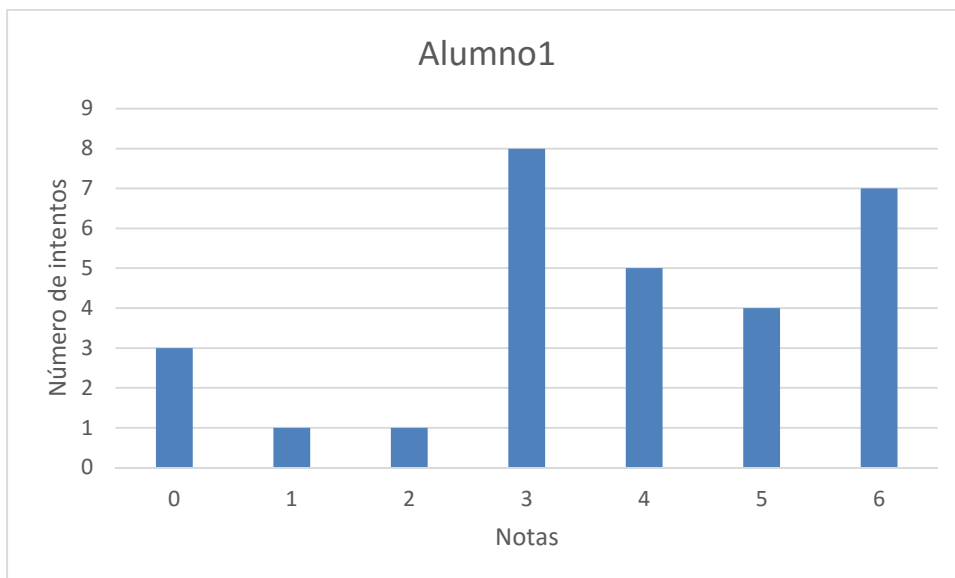


Fig. 5.5 Resultados del alumno1

El alumno6 realizó un total de intentos 22 con el siguiente resultado:

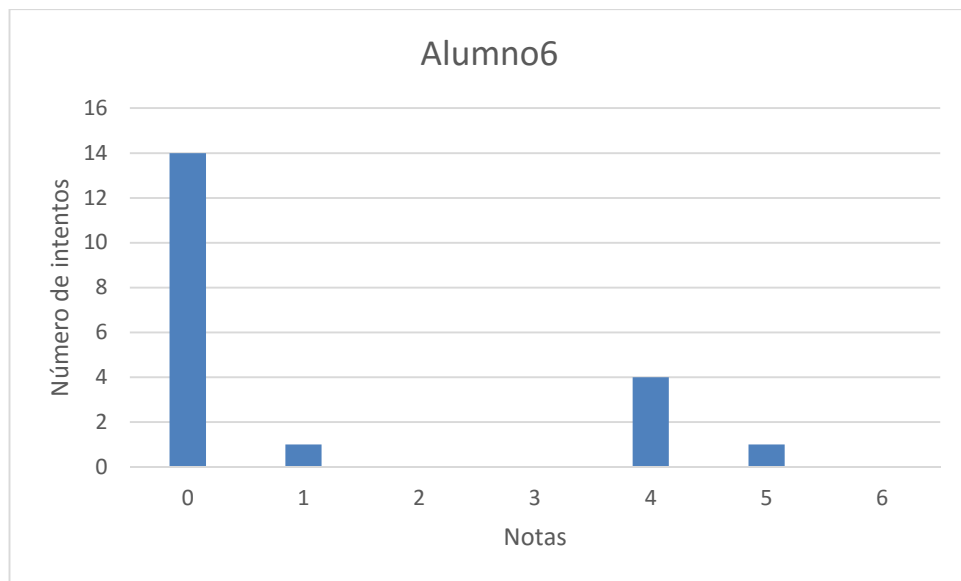


Fig. 5.6 Resultados del alumno6.

5.3 Comparación con notas en ejercicios de clase

Aunque sabemos que la muestra tomada es pequeña, y quizás los datos obtenidos no son todo lo fiables que nos gustasen podemos obtener algunas conclusiones del impacto de la práctica de los alumnos con la aplicación.

Una vez obtenidas las notas de los alumnos en el examen que engloba la convolución en tiempo discreto y continuo, podemos elaborar un histograma de esos resultados.

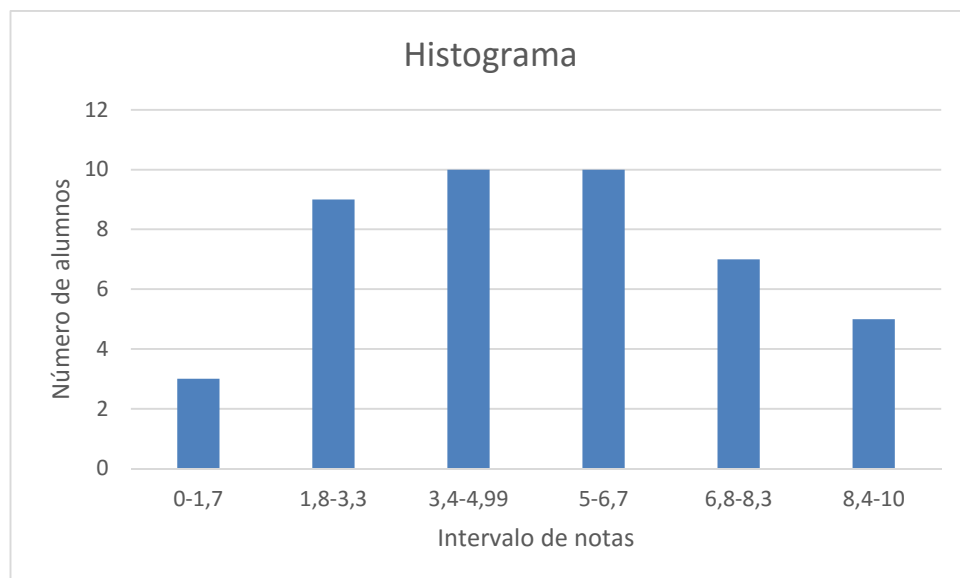


Fig. 5.7 Historia gráfica resultados del examen.

Los datos que obtenemos al ver las notas de los alumnos son:

Alumnos Presentados:44

Alumnos aprobados:22.

Nota media de la clase: 4.96.

Desviación estándar: 2.3.

Los alumnos que elegimos para la muestra obtuvieron la siguiente calificación:

Alumno1: 9.3.

Alumno2: 5.3.

Alumno3: 3.5.

Alumno4: 6.8.

Alumno5: 9.3.

Alumno6: 3.2.

De lo que obtenemos una nota media de: 6.23 y una desviación estándar para este grupo de 2.7.

Teniendo en cuenta que la muestra que tenemos es pequeña, sabemos que no es estadísticamente significativa, pero se aprecia una clara diferencia entre la nota media de los alumnos de la clase y la nota media obtenida por parte de los alumnos que han tomado parte de las pruebas. Esto nos invita a pensar que la aplicación puede tener una buena acogida, y que puede ser útil para el aprendizaje de los alumnos de la asignatura Señales y Sistemas.

6.CONCLUSIONES

Una vez terminado el trabajo, la visión global que tenemos de él nos permite apreciar mejor los resultados obtenidos.

Nos encontramos una aplicación parcialmente desarrollada que supone una buena base sobre la que trabajar. Fuimos dándonos cuenta que esta aplicación permitía muchas mejoras y líneas de evolución.

Nuestro tutor nos emplaza a realizar una serie de modificaciones y mejoras de la aplicación para que esta sea utilizada por los alumnos de la asignatura Señales y Sistemas. Los cuatro principales puntos de trabajo son:

1. Corregir una serie de errores que permite que los enunciados sean legibles y se puedan comprender.
2. La modificación completa del algoritmo que permite el cálculo de la solución de la convolución en el ejercicio continuo, que hasta ahora no era correcta.
3. Conseguir que los datos de la realización de los ejercicios por parte del alumno fueran remitidos al profesor, para que este pudiera hacer un estudio pormenorizado de los errores de los alumnos y así poder enfocar mejor la resolución de dudas de los alumnos.
4. La realización de un nuevo módulo, para que la presentación de los enunciados no sólo sea de manera analítica, sino que también se presente de forma gráfica. La forma en que se determina como se presenta el enunciado responde a un algoritmo, que la presentación del problema sea distinta.

Llegados a este punto podemos afirmar que estamos contentos con los resultados obtenidos.

Hemos cumplido con los objetivos que se nos marcaron al comienzo de este trabajo, los cuatro puntos que hemos nombrado anteriormente están totalmente solventados, y la aplicación funciona correctamente para todas las pruebas a las que la hemos sometido.

Hemos lanzado la aplicación para su uso por parte de los alumnos con un impacto más que satisfactorio sobre las expectativas que teníamos en un principio. Aunque tenemos en cuenta que las pruebas no han sido muy extensas, los resultados nos invitan a ser optimistas y pensar que la aplicación es una buena herramienta para el apoyo del estudio de la asignatura.

A nivel personal, estamos muy satisfechos del trabajo realizado, ya que nos enfrentábamos al desarrollo de una aplicación usando un lenguaje de programación en el cual, por nuestra especialidad de Sistemas electrónicos, no habíamos recibido formación durante la carrera. Terminamos este ciclo sintiendo que estamos preparados para los retos que nos tengan que venir.

REFERENCIAS

EUITT. *PFC Desarrollo de herramientas de autoaprendizaje y evaluación en Señales y Sistemas.* Autor: Ana Osorio, Julio 2011.

Coaten 2003. Coaten, Neil. Blended e-learning. *Educaweb*, 69. 6 de octubre de 2003.

Ceballos 2005a. Ceballos, Francisco Javier. *Java 2. Curso de programación.* Ed Ra-Ma, 2005.

NetBeans. *Documentación herramienta de desarrollo de aplicaciones Java.* Disponible en: (<https://wiki.netbeans.org>)

Java a. *Applets. The Java Tutorials.* Copyright © 1993, 2016, Oracle and/or its affiliates. Disponible en:

(<http://docs.oracle.com/javase/tutorial/deployment/applet/>).

Java b. *Librerías gráficas java.* Copyright © 1993, 2016, Oracle and/or its affiliates. Disponible en:

(<https://docs.oracle.com/javase/7/docs/api/java/awt/Graphics2D.html>)

Java c. Curso de Java. Disponible en:

(<https://www.programarya.com/Cursos/Java/>)

W3C 1998. World Wide Web Consortium (W3C), *Document Object Model (DOM) Level 1 Specification, Version 1.0, W3C Recommendation.* 1 Octubre de 1998. Disponible en: (<http://www.w3.org/TR/REC-DOM-Level-1/>).

ADL 2002. Advanced Distributed Learning Initiative. *Advanced Distributed Learning*

Sharable Content Object Reference Model. Version 1.2. Conformance Requirements, 2002. Disponible en: (www.adlnet.org)

ADL 2004b. *Overview Book. Versión 1.3. Advanced Distributed Learning.* Disponible en: (<http://www.adlnet.org/>).

ADL 2004a. *Content Aggregation Model. Version 1.3.* Disponible en: (<http://www.adlnet.org/>).

ADL 2004c. *Run-Time Environment Model. Versión 1.3. Advanced Distributed Learning.* Disponible en: (<http://www.adlnet.org/>).

ADL 2004d. *Sequencing and Navigation Model. Versión 1.3.* Disponible en: (<http://www.adlnet.org/>).

IMS 2003a. *Content Packaging Information Model. Version 1.1.3, final specification.* IMS Global Learning Consortium, Inc. Disponible en: (www.imsglobal.org).

IMS 2003b. *Simple Sequencing Behavior and Information Model. Version 1.0, final specification.* IMS Global Learning Consortium, Inc. Disponible en: (www.imsglobal.org).

IEEE 2003b. *ECMAScript Application Programming Interface for Content to Runtime Services Communication. Standard for Learning Technology 1484.11.2-2003.*

IEEE 2003a. *Data Model for Content Object Communication. Draft 3 Standard for Learning Technology, P1484.11.1.*

Moodle a. *Moodle Overview.* Disponible en (<https://docs.moodle.org>)

Moodle b. *SCORM.* Disponible en: (https://docs.moodle.org/30/en/SCORM_activity)

