

Universidad de Alcalá  
Escuela Politécnica Superior

**Trabajo Fin de Grado**

Desarrollo de simulador de actividades para aprendizaje basado  
en juego

**Autor:** Irene Dorda Lerín

**Tutor:** Rosa María Estriégana Valdehita

ESCUELA POLITECNICA  
SUPERIOR

**TRIBUNAL:**

**Presidente:** Virginia Escuder Cabañas

**Vocal 1º:** Juana María López Fernández

**Vocal 2º:** Rosa Estriégana Valdehita

## Índice

Resumen	4
Abstract	5
Palabras clave	6
Glosario de acrónimos y abreviaturas	7
Introducción	8
Objetivos y campo de aplicación	9
Medios	10
Herramientas	10
FortiClient	10
FileZilla Client	10
XAMPP	10
phpMyAdmin	10
Visual Studio Code	11
Diagrams.net	11
GNU Gimp	11
Control Web Panel	11
Lenguajes de programación y librerías	11
HTML	11
PHP	11
JavaScript	11
SQL	12
CSS	12
Bootstrap	12
Font Awesome	12
Equipos	12
Ordenador portátil	12
Análisis de requisitos	13
Diseño	15
Ejercicio de Cronogramas	15
Marco teórico	15
Planteamiento final	16
Ejercicio de Minterms y Maxterms	17
Marco teórico	17
Planteamiento final	18
Desarrollo y pruebas	19
Ejercicio de Cronogramas	19
Ejercicio de Minterms y Maxterms	24
Integración con la base de datos	28
Sistema de medallas y puntos	31

## *Desarrollo de simulador de actividades para aprendizaje basado en juego*

Descarga de notas	33
Mejoras y solución de errores	34
Error en estadísticas de curso e históricas	34
Asignación de puntos a las actividades	36
Problemas con los decimales en la descarga de notas	37
Integración de las aplicaciones en el servidor	39
Mejoras y pruebas finales	40
Error en la vuelta al menú de los apartados con simulador de memoria	42
Error con caracteres especiales en el apartado de Vídeos	42
Planificación	44
Presupuesto	46
Conclusiones	47
Futuras mejoras	48
Bibliografía	49
Anexos	50
Anexo 1. cronogramas_RD.php	50
Anexo 2. minmax.php	55
Anexo 3. resultados_min_max.php	59

## Índice de figuras

Figura 1. Tipos de registros según su input y output	15
Figura 2. Registro de desplazamiento bidireccional	16
Figura 3. Tabla de verdad	17
Figura 4. Interfaz de Cronogramas. Versión 1	20
Figura 5. Interfaz de Cronogramas. Versión móvil	23
Figura 6. Interfaz de Minterms y Maxterms	25
Figura 7. Interfaz de Minterms y Maxterms. Versión móvil	27
Figura 8. Estructura de tablas	28
Figura 9. Visualización Medalla de Cronogramas	31
Figura 10. Interfaz de Resultados de Minterms y Maxterms	32
Figura 11. Resultado resumen de Minterms y Maxterms	32
Figura 12. Asignación de puntuaciones. Antes	36
Figura 13. Asignación de puntuaciones. Después	36
Figura 14. Interfaz de CWP	39
Figura 15. Cronograma resuelto	40
Figura 16. Interfaz de Cronogramas. Versión final	41
Figura 17. Menú de videos	43
Figura 18. Diagrama de Gantt	44
Figura 19. Tabla de cálculo de tiempo empleado	46

## Resumen

La asignatura Fundamentos de Tecnología de Computadores (780002) de la Universidad de Alcalá dispone de una aplicación web virtual gamificada que incluye material didáctico y de la que los estudiantes pueden hacer uso para continuar su aprendizaje fuera de las aulas.

El objetivo de este Trabajo de Fin de Grado es aportar más contenido a esta plataforma, con el fin de reflejar prácticamente la totalidad del temario de la asignatura en la página web. Esto se hará con la inclusión de nuevos ejercicios, haciendo uso de tecnologías de software a nivel de cliente y de servidor, procesos de bases de datos y el manejo de diferentes lenguajes de programación.

## **Abstract**

The subject Fundamentals of Computer Technology (780002) at the University of Alcalá has a virtual gamified web application that includes didactic material and that students can use to continue their learning outside the classroom.

The objective of this Final Degree Project is to provide more content to this platform, with the aim of reflect almost the entire content of the subject on the website. This will be done with the inclusion of new exercises, making use of software technologies at client and server level, database processes and the use of different programming languages.

## **Palabras clave**

Online Learning Environment, desarrollo web, aplicación web, gamificación.

## Glosario de acrónimos y abreviaturas

- OLE: Online Learning Environment (Entorno de Aprendizaje en Línea)
- OLEE: Online Learning English Environment (Entorno de Aprendizaje en Inglés en Línea)
- VPN: Virtual Private Network (Red Privada Virtual)
- FTP: File Transfer Protocol (Protocolo de transferencia de archivos)
- XAMPP: X (para cualquier sistema operativo), Apache, MariaDB/MySQL, PHP, Perl
- CWP: Control Web Panel
- URL: Uniform Resource Locator



## Introducción

*Online Learning Enviroment* es una aplicación web didáctica e interactiva desarrollada para la asignatura *Fundamentos de Tecnología de Computadores*, impartida en el primer curso de los grados de Ingeniería Informática e Ingeniería de Computadores de la Universidad de Alcalá, en la que se introducen el funcionamiento y los conocimientos básicos de un computador.

Esta herramienta, a la que llamaremos OLE, incluye principalmente actividades gráficas divididas en los diferentes temas en los que se estructura la asignatura, con el fin de facilitar a los estudiantes la comprensión de los conocimientos impartidos en el aula, además de fomentar su interés e implicación por la asignatura. Por otro lado, OLE también ofrece vídeos explicativos y test de conocimientos adquiridos. La visualización de los videos, y la evaluación de los conocimientos mediante las prácticas y los test, proporcionan una puntuación que se sumará a la nota final de la asignatura. La plataforma permite realizar un seguimiento de las actividades realizadas correctamente y de los puntos adquiridos.

OLE incluye técnicas propias de los juegos y las adapta al ámbito educativo, por ejemplo, concediendo “logros” (medallas) al realizar correctamente un ejercicio de manera consecutiva un número concreto de veces. Esto se conoce como gamificación, y permite aumentar de forma considerable la motivación de los alumnos, su rendimiento y, en consecuencia, el nivel de aprendizaje.

Esta metodología de enseñanza respalda la reestructuración del sistema universitario hacia el Espacio Europeo de Educación Superior (EEES), que establece a los estudiantes como protagonistas de su propio aprendizaje, y sienta un precedente, necesario en los primeros cursos de cualquier estudio, para que los alumnos adquieran un rol activo en su propia formación.

## Objetivos y campo de aplicación

El objetivo de este proyecto es ofrecer técnicas de aprendizaje complementario que sean fieles a los principios de motivación e implicación con la asignatura, gamificación, trabajo autónomo, evaluación inmediata y continua, y afianzamiento de conocimientos de forma on-line.

Para conseguir este objetivo, se implementarán en la aplicación web OLE, y en su versión en inglés OLEE, nuevos ejercicios prácticos que hagan referencia a contenidos que todavía no tengan un espacio en dicha aplicación.

Estas actividades serán interactivas y evaluables, formarán parte del sistema de adquisición de medallas, e incluirán un apartado de seguimiento mediante datos estadísticos para la posterior valoración de los conocimientos adquiridos del estudiante y la mejora del método de aprendizaje.

Por otro lado, se tratará de implementar mejoras y resolver diferentes problemas y fallos en el funcionamiento habitual de la página, ya sean detectados por el administrador de la página, docentes de la asignatura, o durante la realización del mismo proyecto.

Por último, dada la necesidad de migrar OLE y OLEE del dominio [www3.uah.es](http://www3.uah.es) a [www6.uah.es](http://www6.uah.es) por cuestiones ajenas a la coordinación de la asignatura, se hará un seguimiento de dicha migración y se harán las readaptaciones y reconfiguraciones necesarias para que tanto el contenido antiguo como el nuevo funcionen y se visualicen de manera correcta.

## Medios

En este apartado se detallarán las diferentes herramientas usadas para el diseño y desarrollo del proyecto, así como los lenguajes de programación, librerías y equipos empleados.

### *Herramientas*

Primero se nombrarán aquellas herramientas esenciales para el acceso y la gestión del servidor proporcionado por la Universidad de Alcalá, donde se alojan OLE y OLEE, y su copia en local a modo de “servidor de desarrollo”. También el editor de texto y aplicaciones de diseño de recursos para la página web.

#### *FortiClient*

Software que ofrece diferentes funcionalidades, pero la que se usará será la de cliente VPN. La conexión a través de VPN proporciona acceso seguro a la intranet de la UAH y, por lo tanto, a sus servidores.

#### *FileZilla Client*

Cliente FTP multiplataforma, rápido y confiable con muchas funciones útiles y una interfaz gráfica de usuario intuitiva. Se usará para conectarse remotamente al almacenamiento de los servidores de OLE y OLEE, y para poder descargar la totalidad de sus archivos.

#### *XAMPP*

Plataforma de software libre que proporciona todos los componentes básicos necesarios para poder crear (con cualquier sistema operativo) un servidor Apache, su gestión y su correcto funcionamiento. Estos componentes son: servidor FTP para la gestión de archivos, herramienta para la gestión de la base de datos (phpMyAdmin), e intérpretes para los lenguajes PHP y Perl, además de un panel de control donde poder modificar la configuración de estos.

Una vez se tenga una copia de los archivos y la base de datos de OLE y OLEE, se usará XAMPP para crear una copia en local con la que trabajar y hacer pruebas antes de subir las ampliaciones y modificaciones a los servidores de la UAH. En nuestro caso, se hará uso de la versión 5.6.40 (que proporciona la misma versión de PHP).

#### *phpMyAdmin*

phpMyAdmin es una herramienta de software libre destinada a la administración de bases de datos MySQL online. Proporciona una interfaz de usuario con la que hacer las operaciones de uso frecuente (administración tablas, columnas, relaciones, índices, usuarios, permisos, etc.), mientras que también tiene la capacidad de ejecutar directamente cualquier instrucción SQL. Se usará para hacer las adhesiones, modificaciones y consultas necesarias en las bases de datos de OLE y OLEE.

### *Visual Studio Code*

Entorno de desarrollo de código multiplataforma. Proporciona herramientas útiles de depuración y control de versiones, además de numerosas extensiones que poder instalar para que la programación sea mucho más cómoda.

### *Diagrams.net*

Software de dibujo de gráficos multiplataforma gratuito y de código abierto. Originalmente está pensado para crear diagramas de flujo, diagramas UML, organigramas o diagramas de red, aunque en este caso se usará para hacer otro tipo de imágenes que formarán parte del contenido de la página.

### *GNU Gimp*

Programa de edición de imágenes digitales libre y gratuito. Usado para el retoque de ciertas imágenes del proyecto y su documentación.

### *Control Web Panel*

Panel de control online, gratuito e intuitivo para servidores web. Permite gestionar la configuración y los servicios del servidor, aparte de administrar sus archivos y la base de datos a través de phpMyAdmin. Será usado para actualizar los cambios de OLE y OLEE en el servidor propio de la Universidad de Alcalá, una vez los cambios estén funcionando correctamente en la copia local.

### *Lenguajes de programación y librerías*

Los lenguajes de programación y librerías utilizadas serán los que ya se estaban usando en versiones anteriores de OLE y OLEE, continuando con el estilo de trabajo llevado hasta ahora.

### *HTML*

Lenguaje de marcas empleado para elaborar páginas web y que define su contenido: títulos, textos, tablas, imágenes, botones, etc.

### *PHP*

Lenguaje de programación que permite el desarrollo de aplicaciones web, el cual es apto para incrustar el lenguaje HTML. Es ampliamente usado para definir las conexiones entre el cliente y el servidor, por ejemplo, las relaciones con la base de datos.

### *JavaScript*

Lenguaje de programación interpretado, orientado a objetos, funcional y de tipado débil, lo que lo hace muy flexible y perfecto para entornos web donde el código se ejecuta del lado del cliente y del servidor. Se usa para programar la lógica, operaciones, procesos... que van detrás de la interfaz web.

### *SQL*

Lenguaje de dominio específico, diseñado para administrar, e interactuar con la información almacenada en bases de datos relacionales. Es un lenguaje fuertemente tipado que hace uso de un sistema de consultas con las cuales podemos actualizar, borrar e insertar datos, controlar el acceso y definir y redefinir los esquemas de estos datos. En este caso, SQL se empleará empotrado en PHP a través de la extensión mysql().

### *CSS*

Lenguaje de definición de estilos usado para el diseño gráfico de páginas web. Funciona en conjunto con otros lenguajes de programación de forma complementaria a estos y permite cosas como dar color y tamaño a los textos, definir márgenes, ajustar el tamaño de una imagen, definir sus bordes, etc.

### *Bootstrap*

Biblioteca multiplataforma y de código abierto para el diseño de páginas web. Proporciona plantillas con diseños, estilos, imágenes y demás recursos para facilitar el diseño de sistemas web. El uso de Bootstrap era esencial para mantener la estética y el diseño ya presente en OLE y OLEE.

### *Font Awesome*

Conjunto de herramientas de fuentes e íconos basado en CSS. OLE y OLEE usaban Font Awesome por sus iconos, presentes en el menú de las páginas, y se ha mantenido su uso para incluir nuevos iconos.

### *Equipos*

A continuación, se detallan las características usadas del equipo empleado para el desarrollo, pruebas e integración del proyecto.

### *Ordenador portátil*

Equipo personal con Windows 10 como sistema operativo, procesador Intel Core i7-7500U 2.7 GHz, 16 GB de memoria RAM y tarjeta gráfica NVIDIA GeForce 940MX.

## Análisis de requisitos

Como ya se ha comentado, el objetivo principal de este Trabajo es la inclusión de nuevos ejercicios en OLE y en su versión en inglés OLEE, de manera que quede cubierto todo el temario de la asignatura y prácticamente todos los ejercicios prácticos que los estudiantes puedan ver en clase. Hasta ahora cuenta con:

- Ejercicios de Sistemas de Numeración: conversión de binario a decimal, de hexadecimal a decimal, de complemento a 1 a decimal y de complemento a 2 a decimal, y viceversa.
- Ejercicios de mapas de Karnaugh: simplificar una función de 2, 3 o 4 variables, usando minterms o maxterms, y su tabla de verdad.
- Ejercicios de lógica combinacional: tablas de verdad de 3, 4 o 5 variables, análisis de circuitos y síntesis de circuitos.
- Ejercicios de sistemas secuenciales: tablas de verdad y transición de biestables tipo JK, RS, T y D, contadores asíncronos (ascendentes y descendentes) y máquinas de mealy.
- Ejercicios sobre memorias: direccionamiento, longitud de direcciones, localizaciones, operaciones y simulación interactiva.
- Sección de videos de YouTube sobre la asignatura.
- Secciones de visualización de los resultados obtenidos en los ejercicios anteriormente mencionados.
- Sección de administración, sólo disponible con el acceso de un perfil de administrador.
- Sección de descarga de notas, sólo disponible con el acceso de un perfil de administrador.

Algunas de las opciones propuestas para completar las actividades de OLE fueron:

- a. Preguntas sobre la Arquitectura de Von Newman, correspondiente a la introducción de la asignatura.
- b. Ejercicios de minterms y maxterms: dada la forma canónica en minterms o maxterms desarrollar la función original o viceversa.
- c. Ejercicio de cálculo de retardos totales en circuitos con puertas lógicas.
- d. Ejercicios de cronogramas de un registro de desplazamiento: dados unos valores de entrada, determinar el estado del registro pasados X flancos de la señal de reloj.
- e. Ejercicios de contadores síncronos.
- f. Ejercicios de tablas de verdad a partir máquinas de estados de Mealy y Moore.

Después de deliberarlo con la profesora de la asignatura y tutora de este proyecto, se decidió implementar los ejercicios de cronogramas de registro de desplazamiento y los ejercicios de minterms y maxterms por ser considerados los más necesarios para cubrir una parte importante de la asignatura no contemplada de los contenidos de la plataforma.

Estos ejercicios tendrán cómo requisitos fundamentales: seguir el diseño actual de la página web, y que este sea adaptable a diferentes pantallas y dispositivos (ordenadores, tabletas, móviles); que su funcionamiento sea correcto y acorde a los ejercicios prácticos que los estudiantes realizan en clase; estar integrados en la base de datos; que su correcta ejecución proporcione una puntuación modificable desde el apartado de administración; formar parte del sistema de adquisición de medallas; tener control de fechas de exámenes, es decir, que no se pueda sumar puntuación al realizar ejercicios de los que ya se han examinado; incluirse en el apartado “Resultados” para el propio seguimiento de los estudiantes; estar presentes en la archivo descargable de las notas del curso.

Además de los ejercicios a implementar, será muy importante realizar mejoras y corrección de errores a la plataforma. Algunos de estos cambios podrían ser:

- Añadir opciones para que sea más sencilla la gestión de los ejercicios, por ejemplo, poder variar el número de intentos correctos consecutivos que hay que hacer de un ejercicio para conseguir su medalla.
- Solucionar la limitación para asignar una puntuación a los ejercicios.
- Solucionar la visualización de la puntuación de algunos ejercicios a la hora de hacer la descarga de notas.
- Revisar vulnerabilidades, con la posibilidad de actualizar la versión actual de PHP a una superior.
- Solucionar cualquier problema que se pueda detectar o implementar cualquier mejora que se pueda proponer durante la elaboración del proyecto.

Para comenzar a trabajar, se contará con la interfaz OLEE (OLE en su versión en inglés), para que el desarrollo de este proyecto no entre en conflicto con el uso habitual de OLE durante el desarrollo de la asignatura y el curso.

## Diseño

A continuación, se va a hacer un estudio de las bases teóricas que sustentarán cada ejercicio a implementar y, con ello, se definirá la estructura del ejercicio, incluyendo su enunciado, los datos proporcionados, su funcionamiento interno y qué deben hacer los estudiantes para conseguir la puntuación asignada a dicho ejercicio.

### Ejercicio de Cronogramas

#### Marco teórico

Un registro de desplazamiento es un circuito formado por biestables (conectados según el tipo de biestable) que pueden almacenar información binaria y que, además, pueden mover esta información desplazando los bits entre los biestables conectados consecutivamente.

Dependiendo de diferentes criterios, podemos clasificar estos circuitos en diferentes tipos. Según su entrada y su salida de datos:

- Entrada en serie – Salida en serie
- Entrada en serie – Salida en paralelo
- Entrada en paralelo – Salida en serie
- Entrada en paralelo – Salida en paralelo

Se puede ver estos tipos de forma visual en la Figura 1.

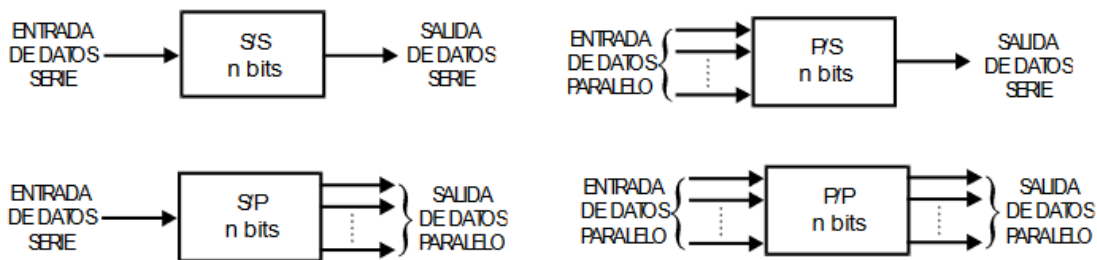


Figura 1. Tipos de registros según su input y output

Según si la salida está conectada a la entrada o no, tendremos un registro de desplazamiento en anillo o un registro de desplazamiento en línea. Además, según el sentido de desplazamiento podrá ser con desplazamiento a la derecha, a la izquierda o bidireccional.

Los registros de desplazamiento bidireccionales son aquellos en que los datos se pueden desplazar a la izquierda o la derecha. Se puede implementar utilizando puertas lógicas que permitan la transferencia de un bit de datos de una etapa a la siguiente, desde la izquierda o desde la derecha, dependiendo del valor de una línea de control. Por ejemplo, si esta señal de control está a nivel alto los bits almacenados en el registro se desplazarán a la derecha, mientras que, si está a nivel bajo, se desplazarán a la izquierda.



Sabiendo todas estas posibilidades, se decide que el elemento principal de este ejercicio será un registro de desplazamiento creado con biestables tipo D, con entrada de datos en serie y salida de datos en paralelo, y bidireccional. Se puede ver la representación de dicho circuito en la Figura 2.

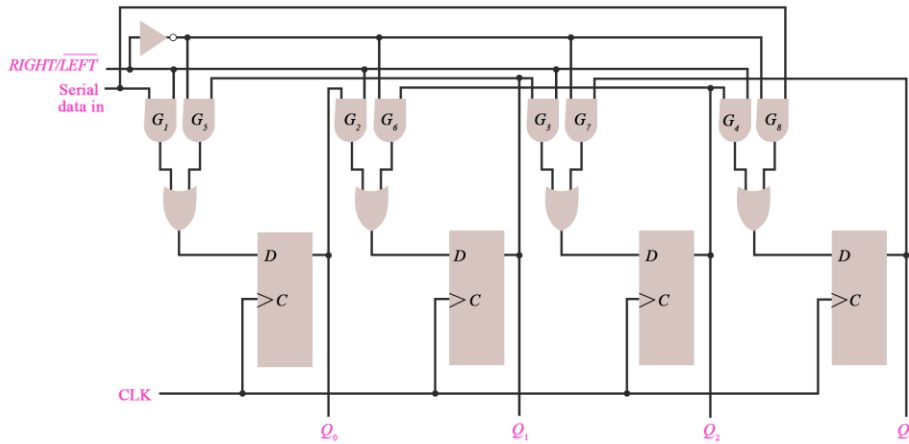


Figura 2. Registro de desplazamiento bidireccional

### Planteamiento final

El ejercicio mostrará la imagen de un cronograma del registro de desplazamiento definido anteriormente. Dicha imagen mostrará los valores de la señal de control del sentido del desplazamiento,  $R/\bar{L}$ ; la señal de reloj, CLK, mostrando si el desplazamiento se ejecuta con flanco de subida o de bajada; la señal de entrada, IN; y los valores iniciales de las salidas,  $Q_0$ ,  $Q_1$ ,  $Q_2$  y  $Q_3$ . Si la señal  $R/\bar{L}$  vale 1, el sentido del desplazamiento será  $IN \rightarrow Q_0 \rightarrow Q_1 \rightarrow Q_2 \rightarrow Q_3$ ; por el contrario, si la señal  $R/\bar{L}$  vale 0, el desplazamiento será  $IN \rightarrow Q_3 \rightarrow Q_2 \rightarrow Q_1 \rightarrow Q_0$ . Los estudiantes deberán ser capaces de completar dicho cronograma y determinar el valor de  $Q_0$ ,  $Q_1$ ,  $Q_2$  y  $Q_3$  pasados 4 flancos de subida o de bajada de CLK.

Para dar aleatoriedad al ejercicio, se creará un banco de 30 imágenes con diferentes cronogramas, 15 que presenten flanco de subida y 15 con flanco de bajada, y que presenten diferentes valores para  $R/\bar{L}$ , IN, y los valores iniciales de las  $Q_s$ .

Cuando al estudiante se le presente un cronograma, tendrá una única oportunidad para dar una respuesta correcta; es decir, tanto si falla como si acierta, el siguiente cronograma siempre será diferente. Para obtener la medalla y la puntuación asociada a este ejercicio, deberá contestar correctamente a 3 cronogramas consecutivos. Este ejercicio formará parte de la Unidad 4 de la asignatura: Sistemas Secuenciales, y, por lo tanto, su acceso se hará desde el mismo menú desplegable que el resto de ejercicios que pertenecen a este tema, y tendrá la misma fecha de examen.

## Ejercicio de Minterms y Maxterms

### Marco teórico

En algebra de Boole se conoce como término canónico aquel en el que aparecen, ya sean sumadas o multiplicadas, todas las variables de una función, ya sea en su forma directa o complementada. Cuando dicho término canónico es una multiplicación de dichas variables, se conoce como *minterm* o minitérmino; cuando se trata de una suma, se conoce como *maxterm* o maxitérmino. Por ejemplo,  $ab'c$  es minterm para una función con las variables  $a$ ,  $b$  y  $c$ .

De la misma manera, se conoce como función canónica a aquella función que está representada con todos sus términos en forma canónica. Cuando la función booleana está expresada en forma de “suma de productos” en realidad es una disyunción de minterms, y cuando se trata de un “producto de sumas” la función es una conjunción de maxterms.

Si tenemos la tabla de verdad de una función lógica  $f(a, b)$  es posible escribir dicha función como suma de minterms o producto de maxterms. Por ejemplo, para la tabla de verdad de la Figura 3:

$a$	$b$	$f(a, b)$
0	0	0
0	1	0
1	0	1
1	1	1

Figura 3. Tabla de verdad

Se puede escribir la función como la suma de los minterms de las filas cuya salida es 1, negando cada variable cuando su entrada es igual a 0:

$$f(a, b) = \sum(2, 3) = ab' + ab$$

Al igual que se puede escribir como el producto de los maxterms de las filas cuya salida es 0, complementando cada variable cuando su entrada sea igual a 1:

$$f(a, b) = \prod(0, 1) = (a + b) * (a + b')$$

Además, aplicando la propiedad distributiva se puede demostrar que ambas funciones son equivalentes:

$$f(a, b) = (a + b) * (a + b') = aa + ab' + ab + bb' = ab' + ab$$

### *Planteamiento final*

En este ejercicio, se presentará una función de 3 variables  $f(a,b,c)$  en forma de sumatorio o productorio, y se pedirá al estudiante que responda con la función canónica en forma de minterms o maxterms. Por ejemplo, para la función  $\Sigma(0,3,7)$  la respuesta sería  $a'b'c' + a'bc + abc$ .

Estas funciones se generarán de forma aleatoria, tanto si es un productorio como un sumatorio, como los números que indican los términos a representar.

Cuando al estudiante se le presente una función, tendrá una única oportunidad para dar una respuesta correcta; es decir, tanto si falla como si acierta, la siguiente función se volverá a generar de forma aleatoria. Para obtener la medalla y la puntuación asociada a este ejercicio, deberá expresar correctamente 3 funciones consecutivas. Este ejercicio formará parte de la Unidad 3 de la asignatura: Lógica Combinacional, y, por lo tanto, tendrá la misma fecha de examen que los ejercicios de este tema, pero tendrá un acceso propio en el menú de la página web.

## Desarrollo y pruebas

Dado que este Trabajo no es un proyecto empezado de 0, sino que se trata de una ampliación y modificación de un proyecto existente, antes de empezar con la programación de los ejercicios, se hará un estudio previo del código de la plataforma para conocer su funcionamiento interno, y que así, a la hora de la implementación, los procesos se ejecuten de la misma manera o similar (adaptada a las nuevas necesidades). Recordemos que el desarrollo inicial se hace en una copia en local de la versión inglesa de OLE.

### *Ejercicio de Cronogramas*

La implementación de los ejercicios comienza con la interfaz de usuario. Se creará un archivo tipo PHP, con código HTML incrustado, para definir la estructura de la página y, por lo tanto, del ejercicio. A este archivo lo llamaremos *cronogramas\_RD.php* y esta será su ruta: *OLEE/secuenciales/cronogramas\_RD.php*.

Para poder acceder a este ejercicio y comprobar así la visualización de su interfaz se modificará el archivo *OLEE/includeMenu.php* para que aparezca con el nombre “Chronograms” con el resto de los ejercicios del tema “Sequential Systems”. El código añadido es:

```
<li><a id="linkMenu31" href=<?php echo $urlRoot . "secuencial/  
cronogramas_RD.php#31"?> >Chronograms</a></li>
```

Haciendo uso de la `<li>` para añadir un elemento nuevo a la lista de ejercicios, y la etiqueta `<a>` que nos permite crear un enlace a la sección de la página web correspondiente a este ejercicio, de manera que cuando se haga click en este elemento, nos lleve a dicho enlace.

El archivo *cronogramas\_RD.php* empezará, como la mayoría de archivos, recibiendo el encabezado de *acceso.php*, en donde se encuentran los assets necesarios para seguir la estética de la plantilla de Bootstrap.

Continuará con la sección de código HTML, empezando por la cabecera `<head>`, en la que referenciaremos el archivo javascript que explicaremos a continuación, y la hoja de estilos *estilo.css* usada también en el resto de ejercicios de Sistemas Secuenciales, y con la que podremos dar formato a los elementos de la página.

Después se define el cuerpo `<body>` del código, donde se especifican los elementos visuales principales de la página. Estos son:

- el título del ejercicio, con etiqueta `<h2>` y valor “Shift-register Chronogram”;
- el enunciado, con etiqueta `<h4>` e id = “enunciado”;
- la imagen del cronograma a completar, con etiqueta `<img>` e id = “imagen”;
- un campo de texto para que el estudiante introduzca su respuesta, con etiqueta `<input>` e id = “respuestaSecuencia”;
- un mensaje de espera, acierto o fallo, con id = “segundo”;

- un mensaje para indicar el número de intentos que lleva el estudiante, con id = “tercero”;
- un botón al que el estudiante tendrá que clickar para comprobar si su respuesta es correcta, con la etiqueta <input> e id = “boton”;
- un mensaje que dice cuál era la respuesta correcta en caso de fallo, con id = “solucionResulta”;
- dos elementos ocultos que nos servirán más adelante en la relación de la página con la base de datos, con id = “resultado” e “id = “correctos”.

Estos elementos estructurados y con el formato de la hoja de estilos referenciada se verán como muestra la Figura 4.

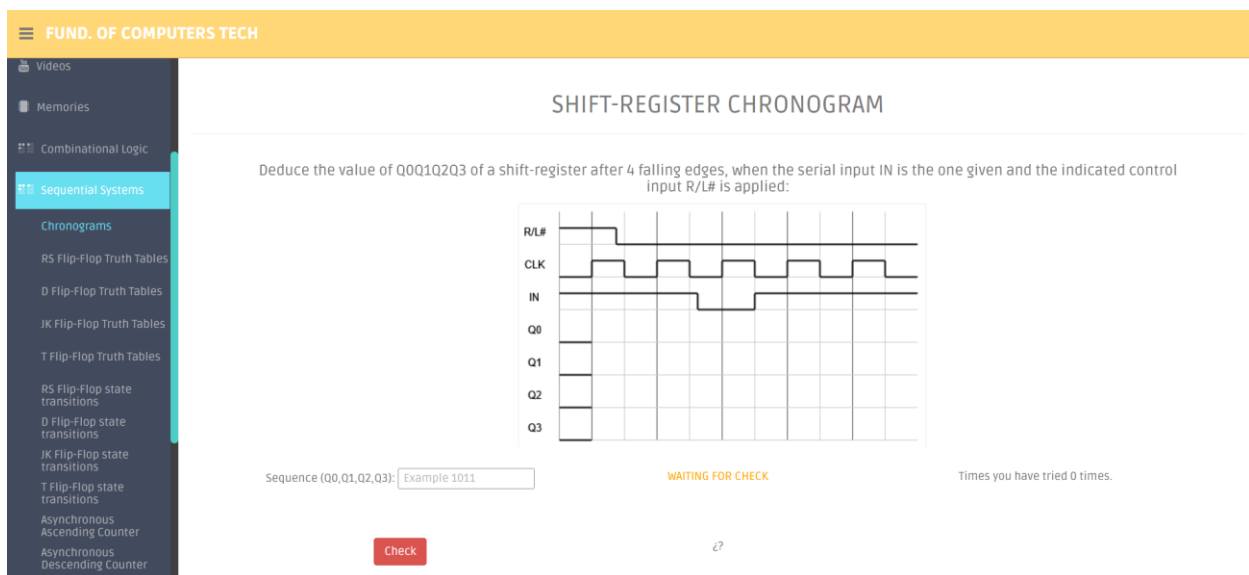


Figura 4. Interfaz de Cronogramas. Versión 1

Las imágenes del cronograma se elaborarán con la herramienta online Diagrams.net, modificando, como señalamos anteriormente, los valores de  $R/\bar{L}$ , IN, el flanco de activación de CLK y los valores iniciales de la  $Q_s$ , quedando al final 30 imágenes distintas a las que llamaremos *crono\_1.png*, *crono\_2.png* ... *crono\_30.png* y que se almacenarán en el directorio *OLEE/secuenciales/src/cronogramas*.

Una vez se tenga la interfaz, el siguiente paso es definir en un documento javascript la lógica interna del ejercicio. A este archivo lo llamaremos *cronogramas\_RD.js* y su ruta será *OLEE/secuenciales/js/cronogramas\_RD.js* y su contenido deberá tener 3 funciones principales: una función de carga de ventana, donde se dará a los elementos unos valores iniciales; una función que seleccione una imagen de cronograma aleatoria del banco de imágenes (recordemos que tendremos 30 imágenes diferentes); y una función que compruebe si la solución dada por el estudiante es correcta o incorrecta.

Antes de definir las funciones, debemos declarar una serie de variables globales que estarán presentes en la mayoría de los procesos:

- *intentos*, inicialmente valdrá 0 y contabilizará el número de intentos que lleva el estudiante.
- *p*, guardará el número del cronograma seleccionado (valdrá de 0 a 29).
- *secuencias[]*, un array que almacene la solución de cada cronograma, en orden.
- *solucionCorrecta*, almacenará la solución correcta del cronograma seleccionado aleatoriamente
- *resultadoCorrecto*, booleano que valdrá false en caso de que la respuesta sea incorrecta, y true en caso de que la respuesta sea correcta.

Con ellas, se podrán realizar los procesos necesarios para el correcto funcionamiento del ejercicio de cronogramas. La función *dibujoAleatorio()* será la encargada de seleccionar aleatoriamente un cronograma. Para ello, hará uso de las funciones matemáticas del objeto incorporado *Math*. Con *Math.random()* se obtendrá un número aleatorio entre 0 (incluido) y 1 (excluido). Este número se multiplicará por la longitud del array *secuencias[]* (en este caso, 30) y el número resultante se redondeará al número entero menor más cercano con *Math.floor()*. De esta manera, obtendremos números aleatorios de 0 a 29, y el resultado se almacenará en *p*. El cambio de imagen se realizará cambiando la ruta del elemento con id = "imagen" de *cronogramas\_RD.php* de esta manera:

```
imagen = document.getElementById("imagen").src =  
"../secuencial/src/cronogramas/crono_" + (p+1) + ".png";
```

En este momento, también se almacenará la solución correcta del cronograma seleccionado igualando *solucionCorrecta* a *secuencias[p]*.

```
solucionCorrecta = secuencias[p];
```

La función *comprobar()* será la que confirme si la respuesta dada es correcta o no. Primero aumentará en 1 el valor de la variable *intentos*, para contabilizar este número. Después, comparará la respuesta del estudiante con la variable *solucionCorrecta*; si el resultado es true, se dará los mensajes y alertas de acierto correspondientes, se almacenará el resultado y la página se preparará para mostrar el siguiente cronograma; si el resultado es false, la alerta será de fallo. Como en este ejercicio sólo se tiene una oportunidad para acertar, al fallar también se almacenará el resultado del intento y se preparará para mostrar otro cronograma aleatorio. A continuación, se puede ver el código de preparación en caso de fallo:

```
/** La variable a intentos se iguala a 1 porque los estudiantes solo
tienen un intento
 * para dar la respuesta correcta. Si fallan, cambia el cronograma.
 */
if (intentos == 1 && !resultadoCorrecto) {
    document.getElementById("solucionResuelta").innerHTML = "La respuesta
correcta es " + solucionCorrecta;
    window.alert("Guardando datos. Lo sentimos, no lo has logrado.");
    intentos = 0;
    document.getElementById("segundo").innerHTML = "ESPERANDO";
    document.getElementById("segundo").className = "col-md-4 esperando";
    document.getElementById("solucionResuelta").innerHTML = "?";
    document.getElementById('resultado').value = intentos;
    document.getElementById('correctos').value = "Último enunciado:
correcto en el " + intentos + " intento.";
    document.getElementById("form1").submit();
}
```

La función *window.onload()* se ejecutará cuando se cargue la ventana, llamando a la función *dibujoAleatorio()* para escoger el cronograma aleatoriamente y preparando el botón de comprobación para que ejecute la función *comprobar()* al clickar sobre él.

```
/**
 * Función de carga de ventana
 */
window.onload = function() {
    var botonComprobar;

    botonComprobar = document.getElementById("boton");
    botonComprobar.onclick = comprobar;
    document.getElementById("segundo").innerHTML = "ESPERANDO";
    document.getElementById("tercero").innerHTML = "Intentos realizados:
" + intentos + ".";
    document.getElementById("solucionResuelta").innerHTML = "?";

    dibujoAleatorio();
}
```

Una vez se tienen estos archivos, se accede a la página web y se hacen las pruebas correspondientes para comprobar la correcta generación aleatoria de los cronogramas, que los mensajes de acierto y fallo se muestran correctamente, y que la comprobación se hace de forma adecuada, pasando dichas pruebas sin importantes modificaciones.

Hay que señalar que una de las pruebas más importantes fue la comprobación de la adaptabilidad del diseño de la web. El navegador usado para este proyecto, Firefox, tiene su propia herramienta de vista de diseño adaptable con la que hacer esta prueba, que fue verificada correctamente, como se puede ver en la Figura 5.

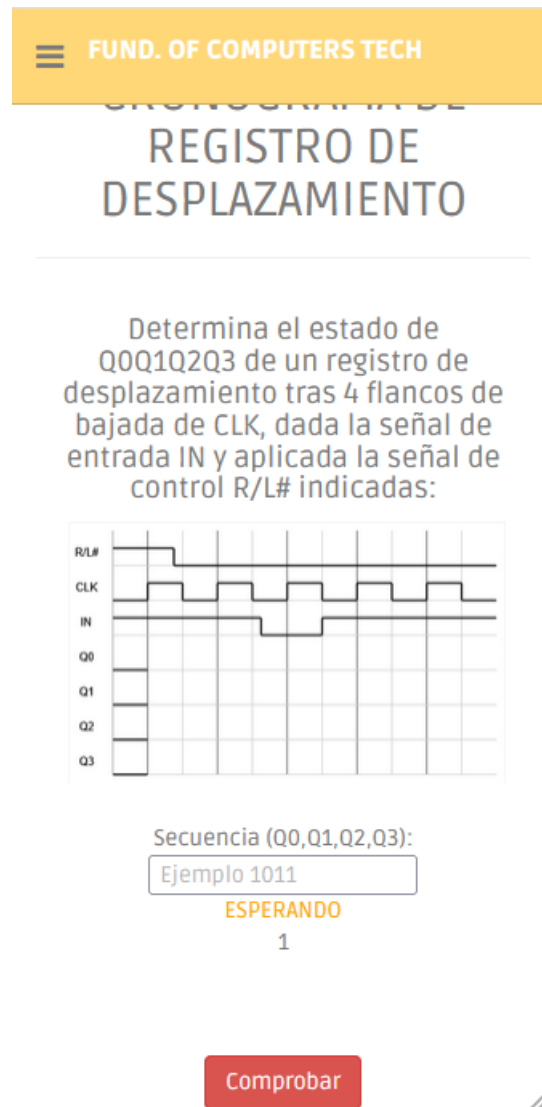


Figura 5. Interfaz de Cronogramas. Versión móvil

Los archivos *cronogramas\_RD.php* y *cronogramas\_RD.js* sufrirán en el futuro algunas modificaciones que explicaremos posteriormente. Sin embargo, se puede ver la totalidad del código de *cronogramas\_RD.php* en el Anexo 1. *cronogramas\_RD.php* de este documento.



## Ejercicio de Minterms y Maxterms

La interfaz de usuario del ejercicio de Minterms y Maxterms, con la que empezará su implementación, se definirá en un archivo llamado *minmax.php* y su ruta será *OLEE/minmax/minmax.php*.

Como se hizo en el ejercicio de Cronogramas, será necesario modificar el menú lateral de OLEE para tener acceso al ejercicio de Minterms y Maxterms, enlazando el nuevo elemento del menú con el archivo *minmax.php*. En este caso, en vez de implementarlo como submenú de algún apartado ya existente, se creará un apartado único para este ejercicio.

```
<li class="sub-menu">
  <a id="linkMenu11" href=?php echo $urlRoot . "minmax/minmax.php#11"?>>
    <i class="fa fa-plus-circle"></i>
    <span>Minterms and Maxterms</span>
  </a>
</li>
```

Como se puede ver, se hace uso de la etiqueta `<i>` para colocar un icono del Font Awesome, en concreto el símbolo + sobre un círculo negro.

El archivo *minmax.php* contendrá la transmisión de la cabecera de *acceso.php* y el código HTML incrustado. En su cabecera, referencias a la hoja de estilos empleada para este ejercicio, *style\_c.css*, usada también en otros ejercicios, como los ejercicios de Numeración; y al archivo javascript que contendrá toda la lógica del ejercicio, y que explicaremos más adelante.

En el cuerpo, se encontrarán los elementos que compondrá la interfaz gráfica, los cuales detallamos a continuación:

- el título del ejercicio, con etiqueta `<h2>` y valor “Minterms and Maxterms”;
- el enunciado, con etiqueta `<h4>`;
- la función a desarrollar, generada aleatoriamente, con `id = “funcion”`;
- un campo de texto para que el estudiante introduzca su respuesta, con etiqueta `<input>` e `id = “respuestaFuncion”`;
- un botón al que el estudiante tendrá que clicar para comprobar si su respuesta es correcta, con la etiqueta `<input>` e `id = “boton”`;
- un mensaje de espera, acierto o fallo, con `id = “solucionResuelta”`;
- un texto a modo de aviso para que los estudiantes no tengan dudas sobre cómo contestar al ejercicio, con la etiqueta `<h6>`;
- dos elementos ocultos que nos servirán más adelante en la relación de la página con la base de datos, con `id = “resultado”` e `id = “correctos”`.

Con estos elementos y la hoja de estilos referenciada, la interfaz del ejercicio de Minterms y Maxterms se verá como ilustra la Figura 6.



Figura 6. Interfaz de Minterms y Maxterms

A continuación, se elaborará el archivo javascript que contendrá los procesos necesarios para que el ejercicio funcione y evalúe las respuestas de los estudiantes de manera correcta. A este archivo le llamaremos *minmax.js* y tendrá la ruta *OLEE/minmax/js/minmax.js*.

Antes de definir las funciones que cumplirán este objetivo, se deben declarar las variables que serán las protagonistas de estas funciones:

- *intentos*, inicialmente valdrá 0 y contabilizará el número de intentos que lleva el estudiante;
- *solucionCorrecta*, almacenará la solución asociada a la función generada;
- *resultadoCorrecto*, booleano que determina si la solución dada por el estudiante es correcta o no;
- *tipo*, variable que determina si la función tendrá que expresarse en Minterms o Maxterms;
- *a*, *b* y *c*, tres variables donde se almacenan los tres términos de la función canónica;
- *min[]* y *max[]*, dos arrays que almacenan la representación con tres variables de los términos canónicos del 0 al 7, en minterms y maxterms respectivamente.

```
min= ["a'b'c'", "a'b'c", "a'bc'", "a'bc", "ab'c'", "ab'c", "abc'", "abc"]
max= ["a+b+c", "a+b+c'", "a+b'+c", "a+b'+c'", "a'+b+c", "a'+b+c'", "a'+b'+c", "a'+b'+c'"]
```

Con estas variables se construirán los procesos internos del ejercicio. Primero, se desarrollará el procedimiento que genera la función matemática a expresar de forma canónica; lo llamaremos *funcionAleatoria()*. El objetivo es generar funciones tipo  $\Sigma(0,3,7)$  o  $\Pi(2,4,5)$ , por lo tanto se debe aleatorizar la producción de 4 elementos: el símbolo de productorio o sumatorio, y los tres números que indican los términos. Para ello, se hará uso de *Math.floor()* y *Math.random()*.

Para el símbolo se usará la variable *tipo*, que valdrá aleatoriamente 0 o 1. Si *tipo* vale 0, la función tendrá que expresarse en suma de minterms y el símbolo será  $\Sigma$ ; si vale 1, la función tendrá que expresarse en producto de maxterms y el símbolo será  $\Pi$ . Para los términos, se usarán las variables *a*, *b* y *c*, que valdrán aleatoriamente valores enteros entre 0 o 7 pero con algunas condiciones. Para que no se nos generen funciones cuyos términos no estén ordenados ascendentemente, o contengan términos repetidos, se configurará para que *a* valga de 0 a 5, *b* valdrá de *a*+1 a 6, y *c* valdrá de *b*+1 a 7. Una vez se tenga el valor de todas estas variables, se igualará el elemento HTML de id = "funcion" a la concatenación de estas, y así se mostrará la función en pantalla.

```
//MINTERMS
document.getElementById('funcion').innerHTML = " $\Sigma$ ("+a+", "+b+", "+c+)";
```

Con este sistema, calcular la expresión canónica de cada función (almacenada en *solucionCorrecta*) será muy fácil, ya que solo habrá que concatenar los términos almacenados en *min[]* o *max[]* de las posiciones de los valores de *a*, *b* y *c*.

```
solucionCorrecta = min[a] + "+" + min[b] + "+" + min[c];
```

Este cálculo se hará en la función *calculoMinterms()*, y habrá una función equivalente para cuando se trate de un producto de sumas, llamada *calculoMaxterms()*.

La función *comprobar()* tendrá una lógica muy similar a la función *comprobar()* del archivo *cronogramas\_RD.js*. Compara la función dada por el estudiante, si es la misma que *solucionCorrecta*, se dará un mensaje de acierto y se almacenará el resultado; si no, se dará un mensaje de error, y como sólo se tiene un intento para contestar bien al enunciado, también se guardará el resultado. Además, en ambos casos, se preparará la página para enseñar una nueva función a expresar canónicamente. A continuación, se puede ver el código ejecutado cuando el resultado es correcto:

```
if (resultadoCorrecto) {
    document.getElementById("resultado").value = intentos;
    document.getElementById("correctos").value = "Último enunciado:
correcto en el " + intentos + " intento.";

    document.getElementById("solucionResuelta").innerHTML = "CORRECTO.
ENHORABUENA";

    window.alert("Guardando. Enhorabuena, respuesta Correcta.");

    document.getElementById("form1").submit();
}
```

Por último, contaremos con la función de carga de página que será la encargada de preparar el botón de la interfaz para ejecutar la función *comprobar()* cuando se pulse, y de llamar al procedimiento *funcionAleatoria()*.

```
/**
 * Función de carga de ventana
 */
window.onload = function() {
  var botonComprobar;

  botonComprobar = document.getElementById("boton");
  botonComprobar.onclick = comprobar;

  funcionAleatoria();
}
```

Una vez se tienen estos archivos, se accede a la página web y se hacen las pruebas correspondientes para comprobar la correcta generación aleatoria de las funciones, la correcta muestra de los mensajes de acierto y fallo, que la comprobación se hace forma adecuada (ya que la respuesta se debe escribir de una forma muy específica) y que el ejercicio posee un diseño adaptable a distintos tamaños de pantalla (como se puede comprobar en la Figura 7), pasando dichas pruebas sin importantes modificaciones.

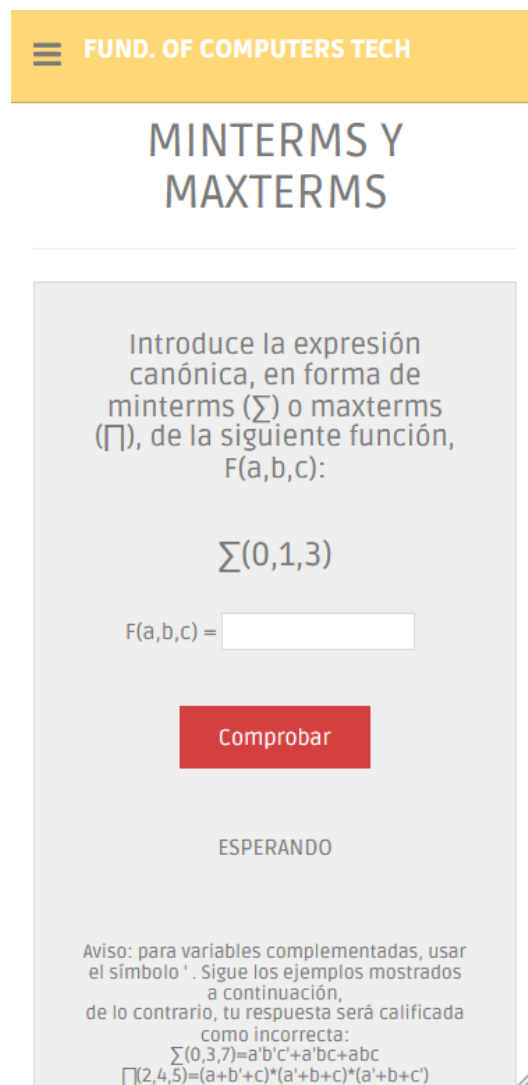


Figura 7. Interfaz de Minterms y Maxterms. Versión móvil

Posteriormente, los archivos *minmax.php* y *minmax.js* surirán unas pequeñas pero necesarias modificaciones esenciales para el desarrollo del proyecto. Sin embargo, se puede consultar la totalidad del código de *minmax.php* en el apartado Anexo 2. *minmax.php* de este documento.

## *Integración con la base de datos*

La integración con la base de datos se refiere a la construcción de las estructuras y de los procesos necesarios para poder almacenar toda la nueva información que surja de los 2 nuevos ejercicios, y poder trabajar con ella más adelante. Analizando la organización de la base de datos y cómo interactúan los demás ejercicios con ella, se llega a la conclusión de la necesidad de hacer algunas modificaciones a la base de datos.

Lo primero que se hará será crear 2 tablas nuevas, una para ejercicio, en la que se almacenará: el número identificador del estudiante que está haciendo el ejercicio, si acierta o falla en su respuesta, y la fecha hora del intento. A estas 2 tablas las llamaremos ‘*tsec\_cronogramas*’ para el ejercicio de Cronogramas, y ‘*tmin\_max*’ para el ejercicio de Mintervals y Maxterms, y ambas tendrán la misma estructura, que se muestra en la Figura 8.

#	Nombre	Tipo
1	<b>num</b> 	bigint(20)
2	<b>resultado</b>	int(11)
3	<b>fecha</b>	timestamp
4	<b>id_usuario</b>	int(11)

*Figura 8. Estructura de tablas*

La columna ‘*num*’ valdrá un número autoincremental y servirá como clave primaria de la tabla; ‘*id\_usuario*’ almacenará el número identificador del alumno que realiza el ejercicio; ‘*fecha*’ almacena de forma predeterminada la fecha y hora del momento en el que se realiza la sentencia SQL al tener la propiedad *CURRENT\_TIMESTAMP*; y ‘*resultado*’ almacena si el intento ha resultado en acierto o en fallo.

Más exactamente, ‘*resultado*’ almacena el número de intentos que le han sido necesarios al estudiante para dar la respuesta correcta, pero cuando falla y no tiene más intentos, los datos se guardan con ‘*resultado*’ igual a 0. Como tanto el ejercicio de Cronogramas como el de Mintervals y Maxterms cuentan con un solo intento para dar una respuesta correcta, ‘*resultado*’ sólo valdrá 0 (fallo) o 1 (acierto).

```
--Cronogramas
CREATE TABLE IF NOT EXISTS `tsec_cronogramas` (
  `num` bigint(20) NOT NULL AUTO_INCREMENT,
  `resultado` int(11) NOT NULL,
  `fecha` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  `id_usuario` int(11) DEFAULT NULL,
  PRIMARY KEY (`num`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--Minterms y maxterms
CREATE TABLE IF NOT EXISTS `tmin_max` (
  `num` bigint(20) NOT NULL AUTO_INCREMENT,
  `resultado` int(11) NOT NULL,
  `fecha` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  `id_usuario` int(11) DEFAULT NULL,
  PRIMARY KEY (`num`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Por otro lado, será necesario modificar la tabla ‘registro’ para añadirle 2 columnas más, una por ejercicio. ‘registro’ es una tabla que contiene toda la información de cada estudiante, desde su nombre, curso y grupo hasta su id, nombre de usuario o contraseña, pero, además, tiene una columna por cada ejercicio que indica si ese estudiante ya ha hecho lo necesario para conseguir la medalla asociada a ese ejercicio o no, normalmente se trata de realizar el ejercicio un número determinado de veces seguidas correctamente. En caso de haberlo hecho, el valor de la columna de ese ejercicio será 1, si no, será 0. se añadirán a ‘registro’ las columnas ‘tsec\_cronogramas’ y ‘tmin\_max’ para los respectivos ejercicios implementados.

```
-- Añadido a registro
ALTER TABLE `registro` ADD `tsec_cronogramas` INT(11) NOT NULL DEFAULT 0;
ALTER TABLE `registro` ADD `tmin_max` INT(11) NOT NULL DEFAULT 0;
```

Una vez hechas estas modificaciones, será necesario añadir a los archivos PHP de los ejercicios todo el código que permite la relación con la base de datos, que se ejecutará cuando se tengan que guardar los resultados. Este código será el mismo que el del resto de ejercicios, pero con sus propios elementos y se puede ver en los Anexos: Anexo 1. cronogramas\_RD.php y Anexo 2. minmax.php de este documento. Se debe recordar que para recibir la medalla o el logro de los ejercicios de Cronogramas y Minterms y Maxterms, se deben realizar 3 veces correctamente y de manera consecutiva.

El código comienza conectando con la base de datos y comprobando si la fecha del examen del tema al que pertenece el ejercicio ya ha pasado, en cuyo caso la ejecución, tanto correcta como incorrecta del ejercicio, no valdrá de nada y no se guardarán los datos. Por el contrario, si el estudiante todavía está en fecha de poder realizarlo antes de la

evaluación de estos contenidos podrá conseguir las medallas y los puntos correspondientes.

En el caso del ejercicio de Cronogramas, el tema es el 4, Sistemas Secuenciales, y en el caso del ejercicio de Minterms y Maxterms, el tema es el 3, Lógica Combinacional.

Si todavía no se tiene la medalla asociada a este ejercicio (comprobando el valor de la columna correspondiente en la tabla 'registro'), el resultado que se va a almacenar actualmente es correcto, y los 2 anteriores son correctos también, significa que con este intento se ha conseguido la medalla y se debe contabilizar en la tabla 'registro', cambiando el valor de la columna correspondiente a 1. Si estas condiciones no se cumplen, simplemente se almacena el resultado de acierto o fallo en la tabla del ejercicio, ya sea 'tmin\_max' o 'tsec\_cronogramas', y se informa al alumno con un mensaje en pantalla.

Una vez se tiene esto completo, se hacen las pruebas necesarias para comprobar la buena interacción entre la página web y la base de datos, con alguna dificultad para contabilizar correctamente la obtención de la medalla en 'registro' al realizar 3 aciertos consecutivos, pero solucionado sin demasiadas complicaciones.

Por último, será necesario hacer una modificación en la tabla 'tblpesos' para añadirle 2 columnas. La tabla 'tblpesos' guarda, para cada curso, la puntuación de cada ejercicio o grupo de ejercicios; por lo tanto, añadiremos las columnas 'sc' y 'mym' para guardar la puntuación que se le dé al ejercicio de Cronogramas y al de Minterms y Maxterms respectivamente.

```
-- Añadido a tabla de pesos
ALTER TABLE `tblpesos` ADD `sc` DECIMAL(2,2) NOT NULL DEFAULT 0.00;
ALTER TABLE `tblpesos` ADD `mym` DECIMAL(2,2) NOT NULL DEFAULT 0.00;
```

Posteriormente habrá que modificar la página web para ser capaz de cambiar el valor de estas columnas desde su interfaz. Esta funcionalidad se encuentra en el apartado de "Administración", accesible solo con el usuario admin y el código que lo permite se encuentra en el archivo *OLEE/admin/cambiapuntuaciones.php*. En este documento se añadirá a la interfaz la caja de texto para introducir la puntuación de los ejercicios de Minterms y Maxterms y de Cronogramas, y se modificará la instrucción SQL para que actualice también los valores de esas columnas. Finalmente se harán las ejecuciones necesarias para comprobar que estos campos se actualizan correctamente.

## Sistema de medallas y puntos

En este apartado se va a explicar cómo se ha preparado la interfaz gráfica de la página para que los estudiantes puedan visualizar su obtención o no de las medallas asociadas a los ejercicios de Minterms y Maxterms, y de Cronogramas, así como su progreso global incluyendo estos ejercicios. Los cambios se centrarán en el menú “Resultados” de la página web, y en el directorio *OLEE/resultados* de sus archivos.

El ejercicio de Cronogramas, al ser un submenú del menú de ejercicios de Sistemas Secuenciales, se incluirá en el subapartado de Resultados de los ejercicios de Sistemas Secuenciales, teniendo que modificar el archivo *OLEE/resultados/resultados\_sec.php*.

A este archivo se añadió la consulta a ‘registro’.’tsec\_cronogramas’, para saber si se ha obtenido la medalla correspondiente a este ejercicio o no, y a ‘tblpesos’.’sc’ para saber la puntuación de dicho ejercicio. El cálculo de la nota se hace multiplicando estos valores, porque si no se tiene la medalla y ‘tsec\_cronogramas’ vale 0, el resultado de la multiplicación será 0; pero si vale 1, el resultado será ‘sc’. Este valor se añade a la nota total de todos los ejercicios de Sistemas Secuenciales.

```
$instruccion6 = "SELECT sc FROM tblpesos WHERE curso = '$fila[2]'";  
$consulta6 = mysql_query($instruccion6, $conexion) or die ("Fallo en la  
consulta 1.6");  
$peso6 = mysql_fetch_row($consulta6);  
$peso_sc = bcddiv($peso6[0],1,3);  
$mCronogramas = $fila[49]*$peso_sc;
```

Siendo \$fila[49] el valor de ‘registro’.’tsec\_cronogramas’.

A su vez, se añade este ejercicio en el cálculo de las estadísticas de curso e históricas de los ejercicios de este tema. Estas estadísticas mostraban el número de estudiantes del mismo curso y de la historia de la plataforma que tuvieran de 0 a 5 medallas. Con la adhesión del nuevo ejercicio, el número posible de medallas cambia a ser de 0 a 6.

Por supuesto, se incluyen estos cambios en la interfaz gráfica de la página y se comprueba que la lógica interna y la visualización (Figura 9) se ejecutan correctamente: que cuando se tenga la medalla de Cronogramas se visualice correctamente, aparezca su puntuación, se sume a la nota final y aparezca en las estadísticas; y en caso contrario, que no se haga.

To get the Chronograms Medal complete the following exercise 3 consecutive times: "Chronograms" (0,050 pt)



Figura 9. Visualización Medalla de Cronogramas

En el caso del ejercicio de Minterms y Maxterms, al tener su propio apartado en el menú, también debe tener su propio subapartado en “Resultados” y, por lo tanto, un único archivo para él, al que llamaremos *resultados\_min\_max.php*.



## Desarrollo de simulador de actividades para aprendizaje basado en juego

En este archivo, con el mismo funcionamiento que el resto de los documentos de “Resultados”, se inicia una conexión con la base de datos para consultar ‘registro’.’tmin\_max’ y ‘tblpesos’.’mym’, y sus valores se multiplican para conocer la nota total. Por otro lado, también se hace el cálculo de las estadísticas históricas y de curso, siendo las opciones tener 0 medallas o 1 medalla.

Por último, se incluye el código HTML que define la interfaz gráfica que nos permitirá visualizar este seguimiento. La Figura 10 muestra el resultado de este desarrollo.



Figura 10. Interfaz de Resultados de Minterms y Maxterms

En el Anexo 3. resultados\_min\_max.php se puede consultar la totalidad del código de este archivo.

Por último, será necesario modificar el subapartado “Resumen” para que aparecieran en él los nuevos ejercicios. Para ello, se modifica el archivo *resultados\_co.php* para incluir la consulta de las columnas ‘tsec\_cronogramas’ y ‘tmin\_max’ de la tabla ‘registro’ y así conocer el estado de las medallas de estos ejercicios. Posteriormente, se modifica el archivo *resultados\_resumen.php* para incluir la consulta de los pesos de los ejercicios de Minterms y Maxterms, y de Cronogramas, y el cálculo de la nota individual de estos ejercicios y su suma a la nota total. Obviamente también se añade a la interfaz la indicación de si se ha conseguido la medalla de estos ejercicios o no, la puntuación obtenida y la puntuación máxima a obtener. En la Figura 11 se puede ver cómo se vería el ejercicio de Minterms y Maxterms al haber obtenido su medalla y que tuviera asociada una puntuación de 0’05 puntos.



Figura 11. Resultado resumen de Minterms y Maxterms

### *Descarga de notas*

El apartado de “Descargas y Seguimiento” es únicamente accesible desde el perfil de administrador. En este perfil tenemos la opción de descargar un archivo csv del desglose de notas de todos los alumnos pertenecientes a un curso concreto, así como sus datos, y la opción de que ese archivo esté aún más detallado, siendo capaces de ver todo el contenido de la tabla ‘registro’, y de cada uno de los vídeos de la plataforma. A cada opción le corresponde un archivo PHP: en el directorio *OLEE/descargas*, *bajar\_notas.php* y *bajar\_todo.php* respectivamente. El objetivo es que en estas descargas aparezca también la puntuación de los ejercicios de Cronogramas y Minterms y Maxterms, y que la suma de la nota final cuente con estos ejercicios, y para ello tendremos que modificar estos archivos.

En ambos documentos se debe: añadir la consulta para obtener el indicador de medalla obtenida para cada ejercicio (de la tabla ‘registro’); añadir la consulta para obtener la puntuación de cada ejercicio (de la tabla ‘tblpesos’); calcular la nota asociada a cada ejercicio (se recuerda que el cálculo de la nota se hace multiplicando estos valores, porque si no se tiene la medalla el valor almacenado en ‘registro’ será 0, y el resultado de la multiplicación será 0; pero si vale 1, el resultado será el valor del peso); incluir en la suma total las dos nuevas notas; y colocar en la salida los nuevos valores.

Tras la modificación se realizan las pruebas correspondientes para asegurar que ambas descargas se hacen de manera correcta, y el nuevo contenido se visualiza en los documentos csv de manera correcta.

Una vez terminada la inclusión de los ejercicios de Cronogramas y de Minterms y Maxterms en la descarga de notas, se puede concluir que dichos ejercicios cumplen todos los requisitos impuestos para este proyecto.

## *Mejoras y solución de errores*

A continuación, se van a explicar las modificaciones que se han realizado durante o tras el desarrollo de los ejercicios, con el fin de solucionar pequeños errores y mejorar funcionalidades de la plataforma.

### *Error en estadísticas de curso e históricas*

Durante la inclusión de la medalla de Cronogramas en el subapartado de Resultados de Ejercicios Secuenciales se detectó un malfuncionamiento en las estadísticas de este, ya que, tanto las estadísticas históricas como del curso no mostraban los estudiantes con x número de medallas correctamente. Estas estadísticas de curso e históricos sobre las medallas conseguidas tienen la finalidad de motivar a los alumnos a realizar los ejercicios al contrastar y comparar el trabajo realizado por sus compañeros.

En primer lugar, fue necesario un estudio minucioso del código existente, en el archivo con ruta *OLEE/resultados/resultados\_sec.php*. Para explicar el error, debemos entender primero como funcionan las estadísticas que se presentan.

Para las estadísticas del curso, OLE analiza las medallas, en este caso, de los ejercicios de Sistemas Secuenciales de todos los estudiantes que pertenezcan al curso del estudiante que las está consultando y, por lo tanto, que tiene iniciada la sesión. Comprueba, a través de la tabla 'registro', si tiene las medallas de los ejercicios de Enunciados de Biestables, la de ejercicios de Transiciones de Biestables, de Contadores Asíncronos Ascendentes, de Contadores Asíncronos Descendentes, de Máquinas de Mealy y de Cronogramas, cuenta el número total de medallas entre esas seis y almacena dicho conteo sumando 1 en la posición de un array que corresponda a ese número de medallas. Es decir, si el resultado de la suma es de 2 medallas, se sumará 1 a la posición 2 del array. De esta manera, para saber cuántos estudiantes del curso tienen 2 medallas, solo habrá que mirar el valor del array para esa posición. El funcionamiento de las estadísticas históricas es el mismo, pero se mira la totalidad de los usuarios independientemente del curso.

El error estaba en que, aunque se estuvieran analizando los usuarios uno por uno, la comprobación de las medallas de Enunciados y Transición de Biestables siempre se hacía mirando las medallas del estudiante que tuviera iniciada la sesión, como si todos los estudiantes tuvieran las mismas medallas que él, resultando en estadísticas distintas según las medallas que tuviera quién las consulta.

Se solucionó cambiando el nombre de una variable en diferentes puntos del código y se comprobó el buen funcionamiento realizando consultas SQL en la base de datos.

Antes de dejar de lado este tema, se detectó el mismo error en el subapartado de Resultados de los ejercicios de Lógica Combinacional, encontrándose el error en la medalla de Tablas de Verdad. Se solucionó y comprobó su resolución de la misma manera. A continuación, se puede ver la comparativa del código de las estadísticas del

curso de los Resultados de los ejercicios de Lógica combinacional antes y después de la solución del error.

- Antes de solucionar el error:

```
// MI CURSO
$instruccion = "SELECT * FROM registro WHERE curso='$fila[2]'";
$consulta = mysql_query($instruccion, $conexion) or die ("Fallo en la
consulta 2");
for($i=0;$i<9;$i++)
    $nota_c_t[$i] = 0;
while ($fila_c = mysql_fetch_row($consulta))
{
    $mtablas_c = $fila[33]*$fila[34]*$fila[35];
    $manalisis_c = $fila_c[36];
    $msintesis_c = $fila_c[37];
    $nota_c = $mtablas_c + $manalisis_c + $msintesis_c;
    $nota_c_t[$nota_c] += 1;
}
```

- Después de solucionar el error, en el cálculo de \$mtablas\_c:

```
// MI CURSO
$instruccion = "SELECT * FROM registro WHERE curso='$fila[2]'";
$consulta = mysql_query($instruccion, $conexion) or die ("Fallo en la
consulta 2");
for($i=0;$i<9;$i++)
    $nota_c_t[$i] = 0;
while ($fila_c = mysql_fetch_row($consulta))
{
    $mtablas_c = $fila_c[33]*$fila_c[34]*$fila_c[35];
    $manalisis_c = $fila_c[36];
    $msintesis_c = $fila_c[37];
    $nota_c = $mtablas_c + $manalisis_c + $msintesis_c;
    $nota_c_t[$nota_c] += 1;
}
```

### Asignación de puntos a las actividades

Por otro lado, se va a explicar un cambio realizado a petición de la profesora de la asignatura y que tiene que ver con el proceso de cambio de puntuación a las actividades. Como se ha explicado anteriormente, en el apartado de administración existe una sección que permite cambiar la puntuación de las actividades para un curso determinado, almacenando los nuevos valores en la tabla ‘tblpesos’. La cuestión es que existía una limitación a la hora de puntuar algunas actividades ya que para unas se podían poner valores con intervalos de 0’01 pero otras en intervalos de 0’1, lo que hacía muy difícil la asignación.

Esto tiene su origen en el archivo con ruta *OLEE/admin/cambiapuntuaciones.php* ya que es dónde se define el intervalo que puede tener cada casilla donde se escribe la puntuación a asignar, en concreto, el valor del atributo “step” de la etiqueta <input> que se usa para cada casilla del formulario. Cambiando el valor de “step” a 0’01 para todas las casillas se soluciona el problema. Podemos ver el cambio en las Figuras 12 (antes) y 13 (después) a continuación:

Elegir curso:	2013/2014
Sistemas de Numeración:	0,40
Minterms y Maxterms	0,15
Mapas de Karnaugh:	0,35
Visualizacion Videos:	0,2
Ejercicios Videos:	0,4

Figura 12. Asignación de puntuaciones. Antes

Elegir curso:	2013/2014
Sistemas de Numeración:	0,40
Minterms y Maxterms	0,15
Mapas de Karnaugh:	0,35
Visualizacion Videos:	0,22
Ejercicios Videos:	0,48

Figura 13. Asignación de puntuaciones. Después

### *Problemas con los decimales en la descarga de notas*

Por último, se va a explicar un cambio que tiene relación con el archivo csv resultante de la descarga de notas, tanto del desglose normal como del que es más detallado. Algunas puntuaciones, por ejemplo, la puntuación de los ejercicios de Minterms y Maxterms, están asociadas a una sola medalla, como es la medalla de Minterms y Maxterms; pero hay otras puntuaciones que están asociadas a varias medallas, por ejemplo, la puntuación de los ejercicios básicos de Memoria está asociada a las medallas de Direccionamiento, Localización y Operaciones de Memoria. De esta manera, para calcular la puntuación que le corresponde un estudiante de estos ejercicios, se divide el peso asociado entre 3 y se multiplica por el número de medallas que haya obtenido. El problema está en que, si el peso que se le da a los ejercicios de memoria no es divisible entre 3, y como se usan solo tres decimales, cuando se tengan las 3 medallas, al volver a hacer la multiplicación, nos quedará un número decimal muy cercano al peso total, pero sin llegar a ser exacto.

Es decir, si el peso de los ejercicios de memoria es de 0'2, el cálculo se hará primero dividiendo 0'2 entre 3, siendo igual a 0'66666... pero guardando solo tres decimales, 0'666, y al multiplicar 0'666 por 3, la nota final será 0'198, que es muy cercano a 0'2, pero no es exacto.

Este problema aparece para todas las puntuaciones que tienen asociada más de una medalla: ejercicios básicos de Memoria y ejercicios de Sistemas de Numeración, y para la puntuación asociada a la Visualización y Ejercicios de los Vídeos.

El objetivo es hacer que cuando se tienen todas las medallas, o se han visto/hecho los ejercicios de todos los vídeos, aparezca en la descarga final la nota exacta correspondiente y no un número decimal aproximado. Para ello, se hará una pequeña modificación en los archivos *bajar\_notas.php* y *bajar\_todo.php* para que detecten, antes, de hacer el cálculo, si cumplen las condiciones para tener la puntuación completa, y así asignarla directamente antes de hacer la multiplicación. A continuación, se puede ver el código antes y después de la solución de este error para los ejercicios básicos de memoria.

- Antes, siendo \$peso\_mb la puntuación ya dividida entre 3 y \$row\_cr[\$ind]['mb'] el número de medallas:

```
//Memorias
$mb=$peso_mb*$row_cr[$ind]['mb'];
```

## *Desarrollo de simulador de actividades para aprendizaje basado en juego*

- Después, comprobando que si `$row_cr[$ind]['mb']` es igual a 3, no haga la multiplicación y que el resultado sea directamente `$pesos[6]` que es la puntuación sin dividir:

```
//Memorias
if ($row_cr[$ind]['mb'] == 3) {
    $mb=$pesos[6];
} else {
    $mb=$peso_mb*$row_cr[$ind]['mb'];
}
```

Este mismo cambio se hizo de la misma manera para la puntuación de los ejercicios de Sistemas de Numeración, Visualización de los Vídeos y Ejercicios de los Vídeos, comprobando su funcionamiento haciendo nuevas descargas y analizando los datos.

## Integración de las aplicaciones en el servidor

El último paso de este proyecto será actualizar las plataformas OLE y OLEE en sus servidores oficiales de la Universidad de Alcalá, en el nuevo dominio [www6.uah.es](http://www6.uah.es), con las modificaciones realizadas en local con la herramienta XAMPP.

Primero se comenzará con OLEE, ya que es la versión local que hemos usado. Posteriormente, cuando se compruebe que en OLEE está todo correcto, se traducirán los textos en inglés a español y se llevarán a OLE.

Para la actualización de OLEE, se iniciará sesión en Control Web Panel con el acceso proporcionado por los Servicios Informáticos de la Universidad. Primero se accederá a phpMyAdmin para la actualización de la base de datos, usando las mismas sentencias SQL que se usaron en la versión local.

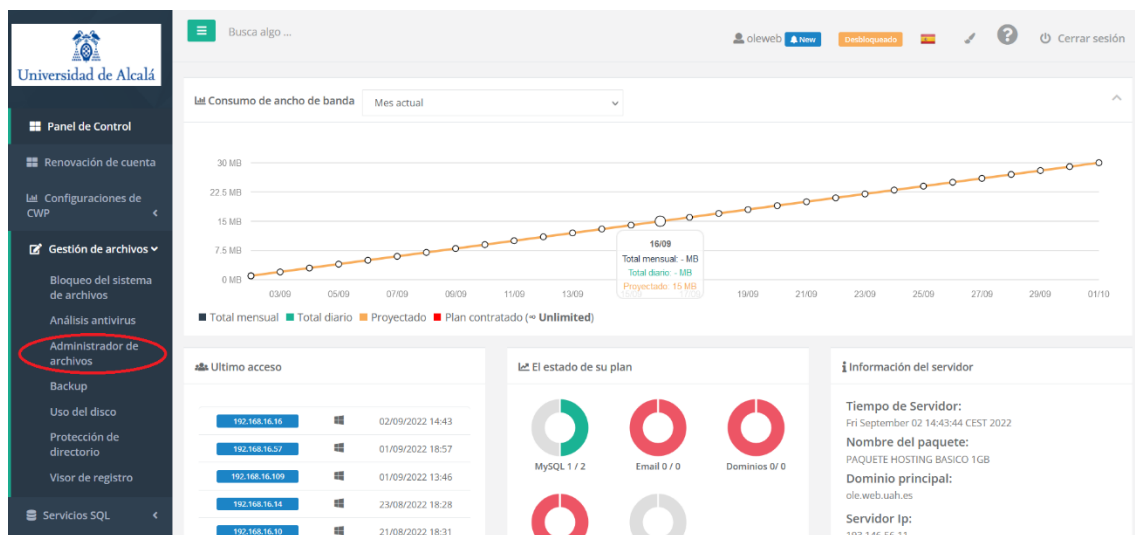


Figura 14. Interfaz de CWP

Posteriormente se accederá a la herramienta de gestión de archivos de CWP (Figura 14) para incluir los nuevos documentos y actualizar los modificados. En conclusión, los documentos de nueva incorporación son:

- *OLEE/minmax/minmax.php*
- *OLEE/minmax/js/minmax.js*
- *OLEE/secuencial/cronogramas\_RD.php*
- *OLEE/secuencial/js/cronogramas\_RD.js*
- *OLEE/secuencial/src/cronogramas* incluyendo las 30 imágenes de los cronogramas
- *OLEE/resultados/resultados\_min\_max.php*

Y los documentos cuyo contenido ha sido modificado son:

- *OLEE/includeMenu.php*
- *OLEE/resultados/resultados\_sec.php*
- *OLEE/resultados/resultados\_co.php*



- *OLEE/resultados/resultados\_resumen.php*
- *OLEE/descargas/bajar\_notas.php*
- *OLEE/descargas/bajar\_todo.php*
- *OLEE/admin/cambiapuntuaciones.php*

Los archivos se fueron subiendo y actualizando paulatinamente para comprobar que todos los procesos iban funcionando correctamente.

### Mejoras y pruebas finales

Una vez todo estaba subido y revisado, la profesora de la asignatura y tutora de este proyecto propuso hacer una modificación del ejercicio de Cronogramas, con el fin de evitar confusión a los estudiantes a la hora de su realización, y para que fuera más semejante a los ejercicios que los alumnos se encuentran en clase. Esta modificación supondría:

- Añadir, en la interfaz del ejercicio, a la izquierda de la imagen del cronograma, el diagrama del registro de desplazamiento bidireccional que se mostró en la Figura 2, modificándolo para que se alternara entre 2 imágenes: una con activación en flanco de subida, y otra con flanco de bajada. Esta edición de imagen se realizó con el programa GNU Gimp.
- Modificar las imágenes de los cronogramas para eliminar información irrelevante y añadir información relevante: eliminar el ciclo sobrante de la imagen, numerar los pulsos, y añadir pequeños elementos visuales. Estas ediciones se realizaron con la herramienta online Diagrams.net. La Figura 15 muestra un ejemplo final de cómo resolverían los estudiantes un cronograma gracias a la nueva información.
- Configurar el enunciado para que indicara si se trataba de flanco de subida o de bajada cuando correspondiera.

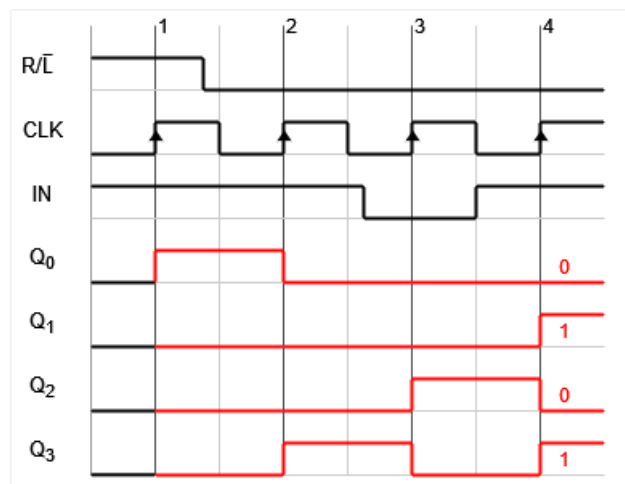


Figura 15. Cronograma resuelto

Para esto, hizo falta rectificar algunos procesos definidos en *cronogramas\_RD.php* y *cronogramas\_RD.js*. Para empezar, en el php, habrá que incluir un nuevo elemento HTML para alojar la imagen del diagrama en la interfaz. Para continuar, en el javascript, habrá que modificar la función *dibujoAleatorio()* para que, antes de seleccionar un cronograma aleatoriamente, determinara aleatoriamente si el cronograma iba a tener flanco de bajada o de subida, y según el resultado, determinar el valor del enunciado, diagrama y cronograma que se correspondiera con ese flanco.

```
switch (flanco) {
  case 0:
    //El cronograma será de flanco de bajada
    document.getElementById("enunciado").innerHTML = "Determina el estado de Q0Q1Q2Q3 de un registro de desplazamiento tras 4 flancos de bajada de CLK, dada la señal de entrada IN y aplicada la señal de control R/L indicadas:";

    //Se asigna la imagen del diagrama
    img = document.getElementById("img").src =
    "../secuencial/src/cronogramas/biestables_bajada.png";

    p = Math.floor(Math.random() * secuencias_bajada.length);
    //Se asigna la solución correcta
    solucionCorrecta = secuencias_bajada[p];
    document.getElementById("solucionResuelta").innerHTML = "?";

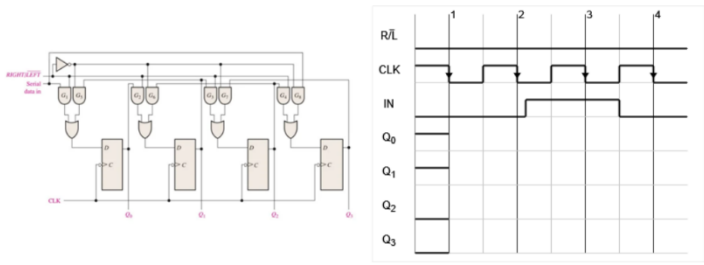
    //Se asigna la imagen del cronograma
    imagen = document.getElementById("imagen").src =
    "../secuencial/src/cronogramas/crono_" + (p+16) + ".png";

    break;
}
```

La Figura 16 muestra nueva interfaz para este ejercicio, frente a la primera versión que vimos en la Figura 4.

### CRONOGRAMA DE REGISTRO DE DESPLAZAMIENTO

Determina el estado de Q0Q1Q2Q3 de un registro de desplazamiento tras 4 flancos de bajada de CLK, dada la señal de entrada IN y aplicada la señal de control R/L indicadas:



Secuencia (Q0,Q1,Q2,Q3):  ESPERANDO Intentos realizados: 0.

[¿?](#)

Figura 16. Interfaz de Cronogramas. Versión final

Una vez dado el visto bueno a estos cambios, se siguió el mismo procedimiento con la plataforma en su versión en español OLE, incluyendo la versión final del ejercicio de Cronogramas. La actualización de OLE se realiza sin ningún tipo de complicación, a través de CWP.

### *Error en la vuelta al menú de los apartados con simulador de memoria*

Se detecta un error en OLE, referente al apartado de “Memoria”. Los subapartados “Simulador Memoria ROM 27C64” y “Simulador Ejercicio Evaluable” poseen una interfaz muy específica donde los elementos son muy grandes y carece de menú lateral para moverse entre apartados. Es por eso por lo que en el lado inferior de estos aparece un botón “volver” que nos lleva atrás y nos permite volver a movernos por el menú. El error residía en este botón, ya que, al pulsarlo, redirigiría a una página de error con el mensaje “file not found”.

Esto se soluciona alterando, en los archivos *simulador\_memoria.php* y *simulador\_evaluable.php*, correspondientes a dichos apartados respectivamente, la función *getUrl()* que pretende proporcionar la url de la página de inicio de OLE para poder volver a ella una vez se pulse el botón “volver”. Una vez hecho esto, la página vuelve a su página de inicio sin problemas.

### *Error con caracteres especiales en el apartado de Vídeos*

Por último, se detecta un error tanto en OLE como en OLEE, surgido a partir de la migración del servidor, por el cual se muestran mal caracteres con acentos o ñ el menú de visualización de “Vídeos”, sustituyéndolos por el símbolo ❖. Para entender por qué ocurre esto únicamente en el título de los vídeos, se debe saber que estos están importados de una tabla de la base de datos, en concreto de ‘tblvideos’, donde los campos que almacenan el tema y el título de los vídeos tienen cotejamiento de tipo “latin1”. El problema está en que OLE y OLEE usan como sistema de codificación “utf-8” la cuál es incompatible con latin1, y por ello muestra problemas con caracteres especiales.

Para solucionar esta diferencia de codificación entre la base de datos y el servidor, vamos a colocar en el código de la plataforma la sentencia `mysql_query("SET NAMES 'utf8'");` que indicará a la base de datos que debe utilizar el conjunto de caracteres “utf-8” para enviar datos al servidor, solucionando así el problema, lo cual podemos comprobar en la Figura 17. Se ha tenido que hacer esta inclusión en todos los archivos referentes al menú lateral de apartado de visualización de vídeos:

- *OLEE/videos/ajax/vv\_menuslat\_vid.php*
- *OLEE/videos/ajax/cv\_menuslat.php*

- [OLEE/videos/ajax/ej\\_refreshmenlat.php](#)
- [OLEE/videos/ajax/fecha\\_menulat.php](#)



**VIDEOS DISPONIBLES**

---

**TEMA 0: Laboratorio**

---

- [Lab.0 MSX88](#)  
Ejercicios del Video: Lab.0 MSX88

---

- [Lab.0 Placa de inserción](#)  
Ejercicios del Video: Lab.0 Placa de inserción

---

**TEMA 1: Introducción**

---

- [Computer History](#)  
Ejercicios del Video: Computer History

---

- [Lab.1 Microinterruptores y Leds](#)  
Ejercicios del Video: Lab.1 Microinterruptores y Leds

---

**TEMA 2: Sistemas de Numeración**

---

- [C1 y C2](#)  
Ejercicios del Video: C1 y C2

---

- [Sistemas de Numeración](#)  
Ejercicios del Video: Sistemas de Numeración

Figura 17. Menú de videos

Con esto, quedan terminadas todas las modificaciones y termina la integración de la plataforma a su servidor oficial.

## Planificación

A continuación, se va a mostrar el diagrama de planificación seguido para este proyecto.

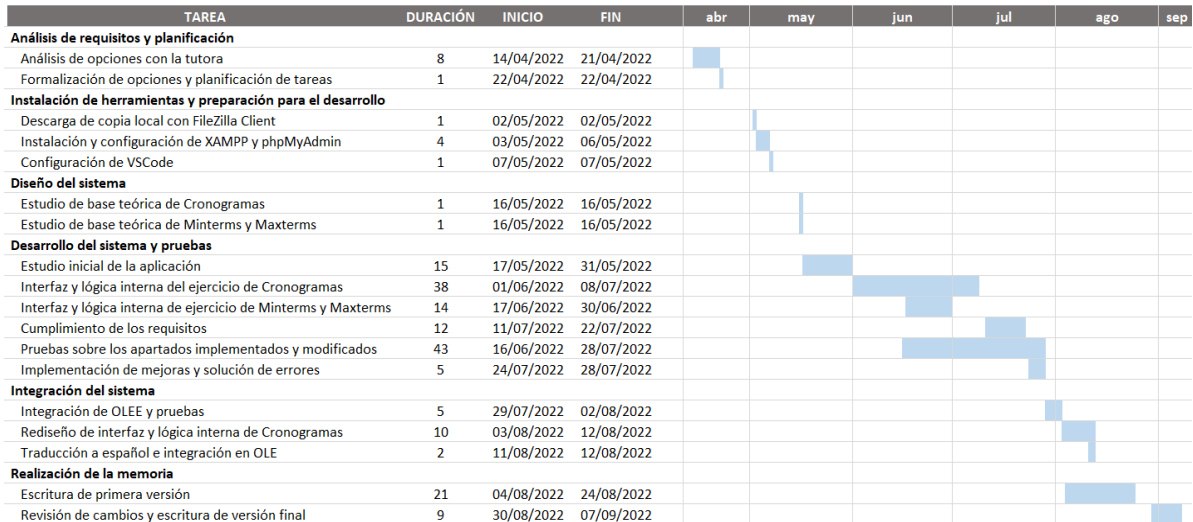


Figura 18. Diagrama de Gantt

Como se puede ver en la Figura 18, las fases de Análisis, Preparación y Diseño fueron relativamente cortas en comparación, debido a la claridad en las propuestas por parte de la tutora y los conocimientos teóricos de los ejercicios a implementar, a excepción de un pequeño bache durante la instalación y configuración de la herramienta XAMPP por incompatibilidades en la versión de PHP.

El trabajo comenzó a alargarse durante la parte de Desarrollo, debido al desconocimiento de la plataforma y de los lenguajes de programación a utilizar. Primero hubo que obtener una primera noción de cada lenguaje (especialmente HTML y PHP) para, con ellos, poder comprender el funcionamiento interno de OLE y comenzar a trabajar, especialmente complejo cuando todos los lenguajes interactúan entre sí.

Esto retrasó el desarrollo más de lo esperado, especialmente en el Ejercicio de Cronogramas, que además sufrió repetidas modificaciones de su interfaz y su lógica interna, tanto en la fase de Desarrollo como en la de Integración. También resultó especialmente difícil toda parte que conlleva relación con la base de datos, muy presente en la tarea Cumplimiento de requisitos.

A pesar de ello, el trabajo en los ejercicios proporcionó la soltura y conocimientos suficientes para que la Implementación de mejoras y solución de errores, y las pruebas y rediseños durante la fase de Integración se realizaran de forma rápida y precisa, incluso siendo capaces de encontrar errores no detectados por la profesora de la asignatura.

Hay que mencionar también que, dado que el Desarrollo se llevó a cabo en varias tareas escalonadas, se trató de realizar pruebas al final de cada una de ellas, con el fin de asegurarse del correcto funcionamiento durante la programación y así no arrastrar errores que provocaran más problemas en el futuro. Es por ello por lo que las Pruebas sobre los

## *Desarrollo de simulador de actividades para aprendizaje basado en juego*

apartados implementados y modificados es una tarea tan larga, porque fue un trabajo continuo y paralelo al desarrollo.

Por último, la fase de Implementación fue bastante sencilla gracias a la intuitiva gestión que permite CWP, a excepción de algunos problemas de detección de cambios en los archivos por parte del servidor web, que se solucionaron rápidamente.

## Presupuesto

Para el cálculo del presupuesto no se han tenido en cuenta costes adicionales como podrían ser el coste de los equipos utilizados, mantenimiento de la base de datos o del servidor web, los costes de los medios usados para el aprendizaje de los lenguajes de programación que usa la plataforma, o los posibles costes en herramientas software (en este caso, despreciables ya que la gran totalidad de las herramientas empleadas son gratuitas y de software libre), por lo que sólo se tiene en cuenta el tiempo de trabajo en el proyecto. Debido a que el tiempo consumido por día se reparte de forma irregular, las horas por cada fase se calculan:

Análisis de requisitos y planificación	4 horas
Instalación de herramientas y preparación para el desarrollo	10 horas
Diseño del sistema	3 horas
Desarrollo del sistema y pruebas	100 horas
Integración del sistema	17 horas
Realización de la memoria	40 horas
Total	174 horas

*Figura 19. Tabla de cálculo de tiempo empleado*

El resultado es un proyecto de 174 horas. Suponiendo un sueldo de 25€/hora al tratarse de una programadora junior, daría un presupuesto final de 4350€ por este Trabajo.

## Conclusiones

El resultado de este trabajo es una plataforma (con versión en inglés y en español) completamente preparada para el inicio del curso 2022/2023. Se ha sido capaz de implementar 2 nuevos ejercicios desde 0 con un funcionamiento óptimo, cumpliendo todos los requisitos puestos por la profesora de la asignatura; corregir diferentes aspectos que impedían hacer un uso adecuado de la aplicación, tanto como usuario regular o como administrador; y asegurar la correcta migración de servidor para OLE y OLEE.

Los estudiantes harán uso de ella para reforzar sus conocimientos, como herramienta para preparar sus pruebas de evaluación y para seguir profundizando en los diferentes aspectos de los Fundamentos de la Tecnología de Computadores. Y para los docentes se convierte en material didáctico muy importante para la enseñanza y un elemento de seguimiento de la evaluación continua.

En concreto, queda fortalecido el entendimiento de los conceptos de “minterm” y “maxterm”, así como el funcionamiento de los cronogramas de un circuito secuencial, esenciales para asignaturas posteriores como *Estructura y Organización de Computadores*.

Para finalizar, a nivel de personal, este Trabajo de Fin de Grado ha supuesto una mejora de las habilidades esenciales para el desarrollo web, siendo estas prácticamente inexistentes al inicio del proyecto. Me ha permitido conocer y manejar con soltura los principales lenguajes de programación del desarrollo web (PHP, HTML, javascript); profundizar en mis conocimientos sobre bases de datos y aplicarlos al manejo de información online; organizar el trabajo individual, estableciendo fases de trabajo, cumpliendo plazos, obteniendo resultados ordenados y acordes a las expectativas; y aplicar técnicas autodidactas, esenciales en el mundo de la programación y de la informática en general.



## Futuras mejoras

Como futuras mejoras se propone, principalmente, la actualización de la versión PHP de la plataforma a una superior. Actualmente, OLE cuenta con la versión PHP 5.6.40, muy lejos de las versiones actuales de PHP 8, y esto hace que el código de la plataforma sea más lento, menos seguro, soporte menos funcionalidades y esté, en definitiva, desactualizado.

Actualmente, para la relación del servidor con la base de datos, OLE hace uso de la extensión mysql, que queda obsoleta a partir de PHP 5.5.0, y eliminada a partir de PHP 7.0.0. En su caso, se deberían utilizar la extensión mejorada mysqli, pero hacerlo supondría tener que cambiar cada línea o fragmento de código que haga uso de mysql, presente una o varias veces en cada uno de los archivos tipo PHP de la plataforma, lo que sería un arduo proceso.

También se propone añadir un nuevo submenú de administración que permita modificar las condiciones necesarias para que los estudiantes puedan adquirir las medallas de los ejercicios, por ejemplo, poder variar el número de intentos correctos consecutivos que hay que hacer de un ejercicio para conseguir su medalla, ya que esos valores están integrados en el código y modificarlos supondría tener que conocerlo, modificarlo, comprobar que se ha realizado el cambio correctamente, etc. Con un submenú esta gestión sería mucho más sencilla.

Por último, dadas las características de la asignatura y de OLE, y aunque en la aplicación se contemplen contenidos (ya sean actividades o vídeos) de cada uno de los temas de la materia, aún se pueden añadir más actividades que reflejen los contenidos de *Fundamentos de Tecnología de Computadores*, por ejemplo, aquellos ejercicios propuestos para su implementación en el apartado Análisis de Requisitos de este documento, siendo estos:

- a. Preguntas sobre la Arquitectura de Von Newman, correspondiente a la introducción de la asignatura.
- b. Ejercicio de cálculo de retardos totales en circuitos con puertas lógicas, como introducción a los circuitos combinacionales.
- c. Ejercicios de contadores síncronos.
- d. Ejercicios de tablas de verdad a partir máquinas de estados de Mealy y Moore.
- e. Cualquier ejercicio que, dada la evolución de los planes de estudios y la metodología de enseñanza, se considere por parte de la profesora de la asignatura.

## Bibliografía

“Fundamentals of Computer Technology”, apuntes de clase, 780002, Departamento de Automática, Universidad de Alcalá, 2021/2022.

L. Welling & L. Thomson, *Desarrollo web con PHP y MySQL* 5ª ed, Madrid: Anaya Multimedia, 2017.

MDN Contributors. Documentación para desarrolladores Web [Online]. Disponible en: <https://developer.mozilla.org/es/docs/Web>

MySQL Documentation [Online]. Disponible en: <https://dev.mysql.com/doc/>

Refsnes Data (1999-2022). W3Schools Web Development Tutorials [Online]. Disponible en: <https://www.w3schools.com/>

The PHP Documentation Group (1997-2022). Manual de PHP [Online]. Disponible en: <https://www.php.net/manual/es/>

## Anexos

Se incluye en este anexo una muestra significativa de el código desarrollado para este proyecto.

### Anexo 1. cronogramas\_RD.php

```
<?php
session_start();
if ($_SESSION["autenticado"] != "SI")
{
    header("Location: ../acceso.php");
    exit();
}
?>

<!DOCTYPE html>
<html lang="es">

<head>

    <title>Cronograma de Registro de Desplazamiento</title>
    <link type="text/css" href="css/estilo.css" rel="Stylesheet"
id="linkestilo" />

    <?php include '../includeCabeceras.php' ?>
    <script src="js\cronogramas_RD.js"></script>
</head>

<body>

    <!--sidebar start-->
    <?php include '../includeMenu.php' ?>
    <!--sidebar end-->

    <section id="main-content">
        <section class="wrapper site-min-height">
            <div class="page-header">
                <!--Titulo visible de la pagina-->
                <h2>Cronograma de Registro de Desplazamiento</h2>
            </div>

            <div class="row mt">
                <div class="col-lg-12">

                    <form id="form1" action="" method="post"
onSubmit="">

                        <div class="container">
```

```
        <!--Enunciado del ejercicio-->
        <h4 id="enunciado">Determina el estado de
Q0Q1Q2Q3 de un registro de desplazamiento tras 4 flancos de bajada de
CLK,
        dada la señal de entrada IN y
aplicada la señal de control R/Ī indicadas:
        </h4>
    </div>

    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <img class="img-responsive"
id="img" alt="Diagrama de Registro de Desplazamiento" src=""
style="margin-bottom: 25px;display: unset;width:37%; height:auto;">
                <img class="img-responsive"
id="imagen" alt="Cronograma de Registro de Desplazamiento" src=""
style="margin-bottom: 25px;display: unset;">
            </div>
        </div>

        <div class="row">
            <div class="col-md-4">Secuencia
(Q0,Q1,Q2,Q3):
                <input type="text"
id="respuestaSecuencia" placeholder=" Ejemplo 1011">
            </div>

            <div id="segundo" class="col-md-4
esperando">ESPERANDO</div>

            <div id="tercero" class="col-md-4
">1</div>

        </div>
    </div>
<br>
    <input type="hidden" id="resultado"
name="resultado">
    <input type="hidden" id="correctos"
name="correctos">

    <br>
    <br>

    <div class="container">
        <div class="row">
            <div class="col-md-4">
```

```

                <input id="boton" type="button"
value="Comprobar" class="btn btn-danger">

                </div>

                <div id="solucionResuelta"
class="col-md-4">¿?</div>
            </div>
        </div>

        <br>

        <!-- AQUÍ VA TODA LA RELACIÓN CON LA BBDD -->
        <?php

            include '../Connections/db.inc';
            if (isset($_POST['resultado']))
            {
                $conexion =
mysql_connect("localhost", $dbuser, $dbpass) or die ("No se puede conectar
con el servidor");

                mysql_select_db($dbname, $conexion) or die
("No se puede seleccionar la base de datos");
                //----- Añadido para consulta de
fechas del temario -----
                $instruccion = "SELECT fecha_examen FROM
tbltemas WHERE tema = '4'";
                $consulta = mysql_query($instruccion,
$conexion) or die ("Fallo en la consulta de fechas de temario");
                $fecha_ex =
mysql_result($consulta, 0); //dejamos la extracción de la fecha del examen
para dar esa información,
                // si esta no fuese necesaria se podrá
quitar o comentar
                echo "Fecha de examen:
$fecha_ex "; //Para comprobar cuando es el examen que lee la consulta

                $instruccion = "SELECT COUNT(tema) FROM
tbltemas WHERE tema = '4' AND CURDATE() > STR_TO_DATE(fecha_examen,
'%d/%m/%Y')";
                $consulta = mysql_query($instruccion,
$conexion) or die ("Fallo en la consulta del número fechas del temario");
                $n_fechas = mysql_result($consulta, 0);

                if ($n_fechas == 0) //Si ha encontrado
fechas previas a la actual es que ya ha pasado el examen
                {
                    //-----
                    -----
                }
            }
        }
    }
}

```

```
        if($_POST['resultado'] != 0)
        {
            $instruccion = "SELECT
tsec_cronogramas FROM registro WHERE id = '$_SESSION[id_usuario]'";
            $consulta =
mysql_query($instruccion, $conexion) or die ("Fallo en la consulta 1");
            $logro =
mysql_result($consulta,0);

            if ($logro == 0)
            {
                $instruccion = "SELECT
resultado FROM tsec_cronogramas WHERE id_usuario =
'$_SESSION[id_usuario]' order by num desc";
                $consulta =
mysql_query($instruccion, $conexion) or die ("Fallo en la consulta 2");
                $nfilas = mysql_num_rows
($consulta);

                if ($nfilas>1)
                {
                    //mira las 2 respuestas
anteriores

                    //como la respuesta
actual es correcta, si las 2 anteriores lo son, se da el logro
                    $anterior0 =
mysql_result($consulta,0);

                    $anterior1 =
mysql_result($consulta,1);

                    if ($anterior0 *
$anterior1)
                        mysql_query("UPDATE
registro SET tsec_cronogramas='1' WHERE id = '$_SESSION[id_usuario]'");
                }
            }
            $instruccion = "INSERT INTO
tsec_cronogramas (id_usuario, fecha, resultado)
VALUES('$_SESSION[id_usuario]',CURRENT_TIMESTAMP,'$_POST[resultado]');
mysql_query($instruccion, $conexion)
or die ("Fallo en la consulta 4");
            mysql_close ($conexion);
            echo $_POST['correctos'];
            //----- Añadido para consulta de
fechas del temario -----
            echo ", Resultado grabado. ";//Para
informar al alumno si se ha grabado su resultado
        } else {
            echo ", fuera de fecha. ";//o está
fuera de fecha y no será grabado
```

```
        }
        //-----
-----
    }

    ?>

</form>

<script>
    dibujoAleatorio();
</script>

</div>
</div>

</section>
</section>

<!--main content end-->

<!--scripts-->
<?php include '../includeFinal.php' ?>

</body>

</html>
```

## Anexo 2. minmax.php

```
<?php
session_start();
if ($_SESSION["autenticado"] != "SI")
{
    header("Location: ../acceso.php");
    exit();
}
?>

<!DOCTYPE html>
<html lang="es">

    <head>

        <title>Minterms y Maxterms</title>
        <link rel="stylesheet" href="../css/style_c.css">
        <?php include '../includeCabeceras.php' ?>
        <script src="js\minmax.js"></script>

    </head>

    <body>

        <!--sidebar start-->
        <?php include '../includeMenu.php' ?>
        <!--sidebar end-->

        <section id="main-content">
            <section class="wrapper site-min-height">
                <div class="page-header">
                    <!--Titulo visible de la pagina-->
                    <h2>Minterms y Maxterms</h2>
                </div>

                <form id="form1" action="" method="post" onSubmit=""
class="basic-grey">
                    <div align="center">

                        <!--Enunciado del ejercicio-->
                        <h4>Introduce la expresión canónica, en forma de
minterms ( $\Sigma$ ) o maxterms ( $\Pi$ ), de la siguiente función, F(a,b,c):</h4>
                        <br>

                        <div id='funcion' style="font-size: 150%" ></div>
                        <br>

                    </div>
                </form>
            </section>
        </section>
    </body>
</html>
```



```
<div>F(a,b,c) =
    <input type="text" id="respuestaFuncion">
</div>
<br>

<input id="boton" type="button"
value="Comprobar" class="button">
<br>
<br>
<br>

<div id="solucionResuelta">ESPERANDO</div>
<br>
<br>

<h6>Aviso: para variables complementadas, usar el
símbolo ' . Sigue los ejemplos mostrados a continuación, <br>
de lo contrario, tu respuesta será calificada
como incorrecta: <br>
    Σ(0,3,7)=a'b'c'+a'bc+abc<br>
    Π(2,4,5)=(a+b'+c)*(a'+b+c)*(a'+b+c')
</h6>

<input type="hidden" id="resultado">
name="resultado">

<input type="hidden" id="correctos">
name="correctos">

<!-- AQUÍ VA TODA LA RELACIÓN CON LA BBDD -->
<?php
include '../Connections/db.inc';
if (isset($_POST['resultado']))
{
    $conexion =
mysql_connect("localhost", $dbuser, $dbpass) or die ("No se puede conectar
con el servidor");

    mysql_select_db($dbname, $conexion) or die
("No se puede seleccionar la base de datos");
    //----- Añadido para consulta de fechas
del temario -----
    $instruccion = "SELECT fecha_examen FROM
tbltemas WHERE tema = '3'";
    $consulta = mysql_query($instruccion,
$conexion) or die ("Fallo en la consulta de fechas de temario");
    $fecha_ex =
mysql_result($consulta, 0); //dejamos la extracción de la fecha del examen
para dar esa información,
```

```

// si esta no fuese necesaria se podrá quitar
o comentar
echo "Fecha del examen: $fecha_ex "; //Para
comprobar cuando es el examen que lee la consulta

$instruccion = "SELECT COUNT(tema) FROM
tbltemas WHERE tema = '3' AND CURDATE() > STR_TO_DATE(fecha_examen,
'%d/%m/%Y')";

$consulta = mysql_query($instruccion,
$conexion) or die ("Fallo en la consulta del número fechas del temario");
$n_fechas = mysql_result($consulta,0);

if ($n_fechas == 0) //Si ha encontrado fechas
previas a la actual es que ya ha pasado el examen
{
//-----
-----

if($_POST['resultado'] != 0)
{
$instruccion = "SELECT tmin_max FROM
registro WHERE id = '$_SESSION[id_usuario]'";
$consulta = mysql_query($instruccion,
$conexion) or die ("Fallo en la consulta 1");
$logro = mysql_result($consulta,0);
if ($logro == 0)
{
$instruccion = "SELECT resultado
FROM tmin_max WHERE id_usuario = '$_SESSION[id_usuario]' order by num
desc";

$consulta =
mysql_query($instruccion, $conexion) or die ("Fallo en la consulta 2");
$nfilas = mysql_num_rows
($consulta);

if ($nfilas>1)
{
//mira las 2 respuestas
anteriores

//como la respuesta actual es
correcta, si las 2 anteriores lo son, se da el logro
$anterior0 =
mysql_result($consulta,0);

$anterior1 =
mysql_result($consulta,1);

if($anterior0 * $anterior1)
mysql_query("UPDATE
registro SET tmin_max='1' WHERE id = '$_SESSION[id_usuario]'");
}
}
}
}
}
```

```
    }
    $instruccion = "INSERT INTO tmin_max
(id_usuario, fecha, resultado)
VALUES('$_SESSION[id_usuario]',CURRENT_TIMESTAMP,'$_POST[resultado]');
mysql_query($instruccion, $conexion) or
die ("Fallo en la consulta 4");
mysql_close ($conexion);
echo $_POST['correctos'];
//----- Añadido para consulta de fechas
del temario -----
echo ", Resultado grabado. ";//Para
informar al alumno si se ha grabado su resultado
} else {
echo ", fuera de fecha. ";//o está fuera
de fecha y no será grabado
}
//----- -----
}
?>

</div>
</form>

<script>
funcionAleatoria();
</script>

</section>
</section>

<!--main content end-->

<!--scripts-->
<?php include '../includeFinal.php' ?>

</body>
</html>
```

### Anexo 3. resultados\_min\_max.php

```
<?php
session_start();
if ($_SESSION["autenticado"] != "SI")
{
    header("Location: ../acceso.php");
    exit();
}
include '../Connections/db.inc';
$conexion = mysql_connect("localhost",$dbuser,$dbpass) or die ("No se
puede conectar con el servidor"); // conexión server
mysql_select_db($dbname,$conexion) or die ("No se puede seleccionar la
base de datos"); // conexión BD
$instruccion = "SELECT * FROM registro WHERE id =
'$_SESSION[id_usuario]'";
$consulta = mysql_query($instruccion, $conexion) or die ("Fallo en la
consulta 1");
$fila = mysql_fetch_row($consulta);

$instruccion1 = "SELECT mym FROM tblpesos WHERE curso = '$fila[2]'";
$consulta1 = mysql_query($instruccion1, $conexion) or die ("Fallo en la
consulta 1.1");
$peso1 = mysql_fetch_row($consulta1);
$peso_mym=bcdiv($peso1[0],1,3);
$mMinMax = $fila[50]*$peso_mym;

$nota = $mMinMax;

//Estadísticas del curso
$instruccion2 = "SELECT * FROM registro WHERE curso='$fila[2]'";
$consulta2 = mysql_query($instruccion2, $conexion) or die ("Fallo en la
consulta 2");
for($i=0;$i<2;$i++)
    $nota_c_t[$i] = 0; // array de 2 elementos (0 medallas y 1 medalla)

while ($fila_c = mysql_fetch_row($consulta2))
{
    $mMinMax_c = $fila_c[50];

    $nota_c = $mMinMax_c; //valores posibles: 0 o 1 (correspondiente al
nº de medallas)
    $nota_c_t[$nota_c] += 1; //lo que hace es calcular el numero de
medallas y sumar 1 a la posición del array correspondiente a ese número
}

//Estadísticas historicas
$instruccion3 = "SELECT * FROM registro";
```

```
$consulta3 = mysql_query($instruccion3, $conexion) or die ("Fallo en la
consulta 3");
for($i=0;$i<2;$i++)
    $nota_h_t[$i] = 0; // array de 2 elementos (0 medallas y 1 medalla)

while ($fila_h = mysql_fetch_row($consulta3))
{
    $mMinMax_h = $fila_h[50];

    $nota_h = $mMinMax_h; //valores posibles: 0 o 1 (correspondiente al
nº de medallas)
    $nota_h_t[$nota_h] += 1; //lo que hace es calcular el numero de
medallas y sumar 1 a la posición del array correspondiente a ese número
}

mysql_close($conexion);

?>
```

```
<html>
```

```
<head>
```

```
<?php include '../includeCabeceras.php' ?>
<link rel="stylesheet" href="../css/style_c.css">
<title>Resultados de Minterms y Maxterms</title>
```

```
</head>
```

```
<body>
```

```
<!--sidebar start-->
<?php include '../includeMenu.php' ?>
<!--sidebar end-->
```

```
<!--
```

```
*****
*****
*****
```

```
MAIN CONTENT
```

```
*****
*****
***** -->
```

```
<!--main content start-->
```

```
<section id="main-content">
```

```
<section class="wrapper site-min-height">
```

```
<div class="page-header">
```

```
<h2>Resultados de Minterms y Maxterms</h2>
```

```
</div>
```

```
<div class="row mt">
```

```
<div class="col-lg-12">
```

```
<form class="basic-grey" style="background:
none;border: none;">
    <div align=center>
        <br>
        <table class="table table-bordered table-
striped">
            <tr>
                <td>
                    Para conseguir la <b>Medalla
de Minterms y Maxterms</b> completa 3 veces consecutivas el siguiente
ejercicio: "Minterms y Maxterms"
                    (<?php echo
number_format($peso_mym,3,"",".") ?> ptos)
                    <!-- To get the <b> Minterms
and Maxterms Medal </b> complete the following exercise 3 consecutive
times: "Minterms and Maxterms"
                    (<?php echo
number_format($peso_mym,3,"",".") ?> pt) -->
                </td>
                <td align=center>
                    <?php if($mMinMax) echo "<img
src='../images/medalla.png' data-toggle='tooltip' title='Achieved medal'
width=35 height=50>";else echo "<img src='../images/no_medalla.png' data-
toggle='tooltip' title='NOT achieved medal' width=35 height=50>"?>
                </td>
            </tr>
        </table>
        <br>
        <br>
        <table class="table table-condensed
table-bordered">
            <tr>
                <td>
                    Puntos totales
                </td>
                <td>
                    <?php echo
number_format($nota,3,"",".")?>
                </td>
            </tr>
        </table>
        <br>
        <br>
        <table class="table table-bordered">
            <tr>
                <td align=center
colspan=4>Estadísticas</td>
```

```

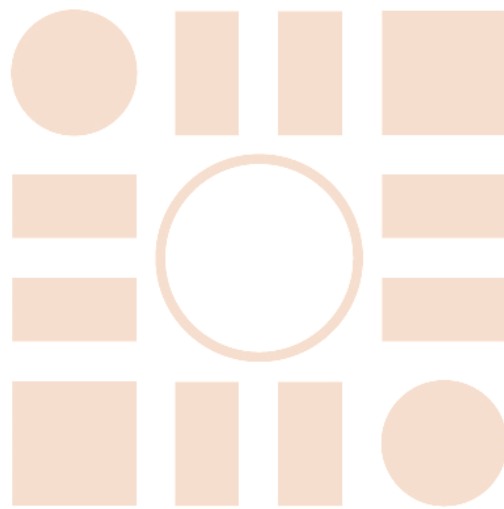
        </tr>
        <tr>
            <td align=center colspan=2>Mi
curso</td>
            <td align=center
colspan=2>Histórico</td>
        </tr>
        <tr>
            <td>0 Medallas</td>
            <td>
                <?php echo $nota_c_t[0]?>
            </td>
            <td>0 Medallas</td>
            <td>
                <?php echo $nota_h_t[0]?>
            </td>
        </tr>
        <tr>
            <td>1 Medalla</td>
            <td>
                <?php echo $nota_c_t[1]?>
            </td>
            <td>1 Medalla</td>
            <td>
                <?php echo $nota_h_t[1]?>
            </td>
        </tr>
    </table>
    <br>
    <br>
</div>
</form>
</div>
</div>
</section>
</section>

<!--main content end-->

<!--scripts-->
<?php include '../includeFinal.php' ?>

</body>
</html>
```

Universidad de Alcalá  
Escuela Politécnica Superior



ESCUELA POLITECNICA  
SUPERIOR



Universidad  
de Alcalá