



Universidad Politécnica de Valencia
Escuela Técnica Superior de Ingeniería Informática

PROYECTO FINAL DE CARRERA

DISCA-10. CONSTRUCCIÓN DE UN SITIO WEB PARA LA REALIZACIÓN DE
EJERCICIOS DE REPRESENTACIÓN DE NÚMEROS ENTEROS

Memoria del Proyecto Fin de Carrera
Ingeniería Informática de Gestión
Realizado por: Jose Sánchez Baviera
Director: Antonio Martí Campoy
Valencia, 2012

ÍNDICE

1	<i>Introducción general</i>	5
1.1	Introducción	5
1.2	Motivación y objetivos	5
1.3	Estructura del documento	6
2	<i>Especificación de requisitos</i>	7
2.1	Introducción	7
2.2	Información inicial	7
2.3	Estimación de recursos	7
2.4	Requisitos para ser usuario registrado	7
3	<i>Descripción del proyecto</i>	9
3.1	Descripción del entorno general del proyecto	9
3.2	Funciones de la aplicación	10
4	<i>Análisis</i>	11
4.1	Casos de Uso	11
5	<i>Diseño</i>	13
5.1	Capa de presentación	13
5.2	Capa de negocio	17
5.3	Capa de datos	18
6	<i>Implementación</i>	21
6.1	BLOQUE I: Login	21
6.2	BLOQUE II: Presentación de ejercicios	23
6.2.1	Lógica de los ejercicios	24
6.3	BLOQUE III: Envío de resultados	26
6.3.1	Encriptación	26
6.3.2	Desencriptación	27
7	<i>Evaluación</i>	29
7.1	Prueba de validación CSS	29
7.2	Prueba de validación sobre diferentes navegadores	29
8	<i>Conclusiones</i>	33
8.1	Valoración personal	33
8.2	Posibles extensiones del proyecto	33
9	<i>Bibliografía</i>	33
10	<i>Anexos</i>	35
10.1	Preparación entorno de trabajo	35
10.2	Configuración base de datos	36

1 Introducción general

1.1 Introducción

El presente documento es la memoria descriptiva sobre el desarrollo de una aplicación Web para la realización de ejercicios de representación de números enteros en la Universidad Politécnica de Valencia, según las especificaciones del proyecto con código DISCA-10, tutorizado por Antonio Martí Campoy.

1.2 Motivación y objetivos

Una opción diferente y cada vez más demandada son los cursos online que permiten al alumno planificar la realización de estos, según su disponibilidad de tiempo, a un ritmo de trabajo independiente y acorde con la capacidad del alumno.

Con esta aplicación, el usuario puede practicar varios tipos de ejercicios, sin estar limitado a un número fijo de estos, ya que la aplicación genera nuevos ejercicios aleatoriamente en cada ejecución, y el alumno puede comprobar los resultados al momento. Además del carácter puramente didáctico, puede servir al profesor para obtener un seguimiento sobre los avances del alumno.

El objetivo principal del proyecto es realizar una aplicación Web para que el alumno pueda practicar y perfeccionar los ejercicios sobre representación de números enteros. Convertirlos a Signo y Magnitud, a Complemento a 1, a Complemento a 2, a Exceso a Z, y poder enviar sus resultados al tutor para la evaluación de éste. Los únicos requerimientos para el alumno serían un navegador Web con conexión a Internet, y ganas de aprender o mejorar la representación de números enteros.

1.3 Estructura del documento

Para presentar la memoria, se ha seguido el estándar IEEE std 830-1998.

- 1. Introducción general**
- 2. Especificación de requisitos**
- 3. Descripción del proyecto**
- 4. Análisis**
- 5. Diseño**
- 6. Implementación**
- 7. Evaluación**
- 8. Conclusiones**
- 9. Bibliografía**
- 10. Anexos**

2 Especificación de requisitos

2.1 Introducción

Este apartado va a explicar el funcionamiento de la aplicación según el estándar IEEE std 830-1998.

2.2 Información inicial

La aplicación presentada permite, de manera remota e independiente, que el alumno pueda practicar y perfeccionar ejercicios de representación de números enteros en diferentes representaciones. La finalidad es que el tutor pueda llevar un seguimiento de los progresos del alumno, así como del tiempo empleado.

2.3 Estimación de recursos

Por la naturaleza de la aplicación, no será necesario ningún tipo de software especial, simplemente un navegador que permita ejecutar código Javascript y una conexión a Internet. Gracias a esta flexibilidad, el alumno tiene completa libertad para realizar los ejercicios en el lugar y momento que considere oportuno sin ninguna restricción.

2.4 Requisitos para ser usuario registrado

El requisito fundamental para poder acceder a la aplicación es estar registrado en la base de datos. Este paso es necesario para que el tutor pueda llevar un seguimiento personalizado de cada alumno. Para ello, el usuario tendrá que rellenar un formulario especificando los siguientes datos:

- DNI: La aplicación usará este valor como el campo identificativo principal, de tal manera que para el alumno, su DNI será su usuario. Este campo tiene que ser único y correcto. En el momento de darse de alta en la aplicación, el formulario chequea que el DNI introducido es correcto (cálculo de la letra de control), y que no existe previamente (si alguna de estas dos condiciones no se cumple, no permite el registro).
- Contraseña: Palabra clave para validar el login, que el alumno introduce sin restricción, necesaria para entrar en la aplicación en sucesivas ocasiones.
- Apellidos: del alumno, necesario para que el tutor pueda identificarlo.
- Nombre: del alumno, necesario para que el tutor pueda identificarlo.
- Correo electrónico: Este campo es imprescindible para que el tutor pueda contactar con el alumno en caso de duda o recuperación de la clave. El formulario genera un pequeño chequeo para asegurar que el correo electrónico tiene la estructura correcta.
- Grupo: Necesario para que el tutor pueda identificar en qué grupo está matriculado.

3 Descripción del proyecto

3.1 Descripción del entorno general del proyecto

Primero de todo, el alumno tiene que ser un **usuario registrado** para poder acceder a la aplicación. Este punto es capital, ya que el tutor requiere un seguimiento de sus resultados, y es la forma que existe de salvaguardar la identidad y la fiabilidad de los resultados. Debe rellenar un formulario para el login, y en caso de que no estuviera registrado previamente, debe darse de alta rellenando unos datos mínimos de identificación.

Es una aplicación meramente didáctica, lo que implica que el alumno es libre de realizar el número de ejercicios que crea conveniente en el tiempo que considere oportuno. Aún así, para el correcto aprovechamiento de la aplicación, sería conveniente que el tutor diera unas pautas mínimas de uso para que fuera útil en la evaluación y no dependiera del criterio de cada alumno:

- Un mínimo de ejercicios realizados
- Un mínimo de tipos de ejercicios trabajados
- Un mínimo de tiempo empleado en la aplicación

Una vez finalizado el periodo de trabajo, el alumno puede optar por enviar los resultados al tutor. Este envío no es obligatorio ni automático, y permite al alumno practicar ejercicios meramente por el perfeccionamiento de estos.

Existen dos formas de enviar los resultados:

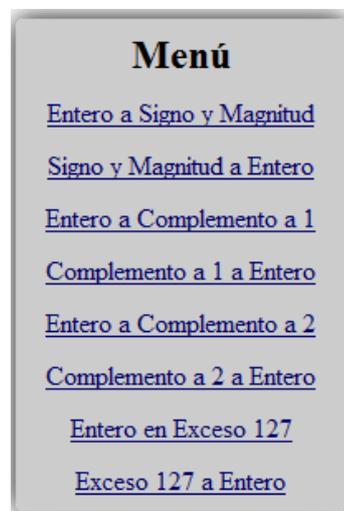
- Mediante un fichero txt. Este fichero contendrá todos los datos del alumno (introducidos en el registro) así como los resultados obtenidos por tipo de ejercicio
- Envío directo a la tabla “ejercicios” de la base de datos creada llamada “pfc”. Para poder identificar al alumno que ha realizado los ejercicios, la tabla “ejercicios” contiene el campo “dni”. Este campo se puede cruzar con el campo “dni” de la tabla “alumnos”, que es único, y poder obtener los datos personales del alumno en cuestión.

En ambas posibilidades, los datos sufren una pequeña encriptación antes de ser enviados al servidor, y una vez allí, son descriptados. Esto permite comprobar que los resultados no han sido vulnerados ni modificados en su envío. Existe un campo “firma” en la tabla “ejercicios”, donde el tutor puede confirmar que el envío ha sido correcto y los valores a evaluar no han sido modificados.

3.2 Funciones de la aplicación

Una vez el alumno está correctamente registrado y logeado, la aplicación le permite realizar los siguientes tipos de ejercicios:

- Representar un número entero en signo y magnitud.
- Representar un número en signo y magnitud en Entero
- Representar un número entero en Complemento a 1, utilizando 8 bits
- Representar un número en Complemento a 1 (representado con 8 bits) en Entero
- Representar un número entero en Complemento a 2, utilizando 8 bits
- Representar un número en Complemento a 2 (representado con 8 bits) en Entero
- Representar un número entero en Exceso 127, utilizando 8 bits
- Representar un número en Exceso 127 (representado con 8 bits) en Entero



Los resultados, así como el tiempo empleado se van actualizando en tiempo real en la tabla que se encuentra en el pie de la página principal. Nos permite ver el número de ejercicios realizados, el número de aciertos, el número de errores, el tiempo empleados, y los datos personales del alumno (Apellidos, Nombre, Mail y Grupo)

Ejercicios: 11	Aciertos: 3	Errores: 8	Tiempo: 1562
Apellidos:	Sanchez Baviera	Nombre:	Jose
Mail:	josanba5@ei.upv.es	Grupo:	tardes
<input type="button" value="EnviarDatos"/>		<input type="button" value="EnviarDatosSQL"/>	

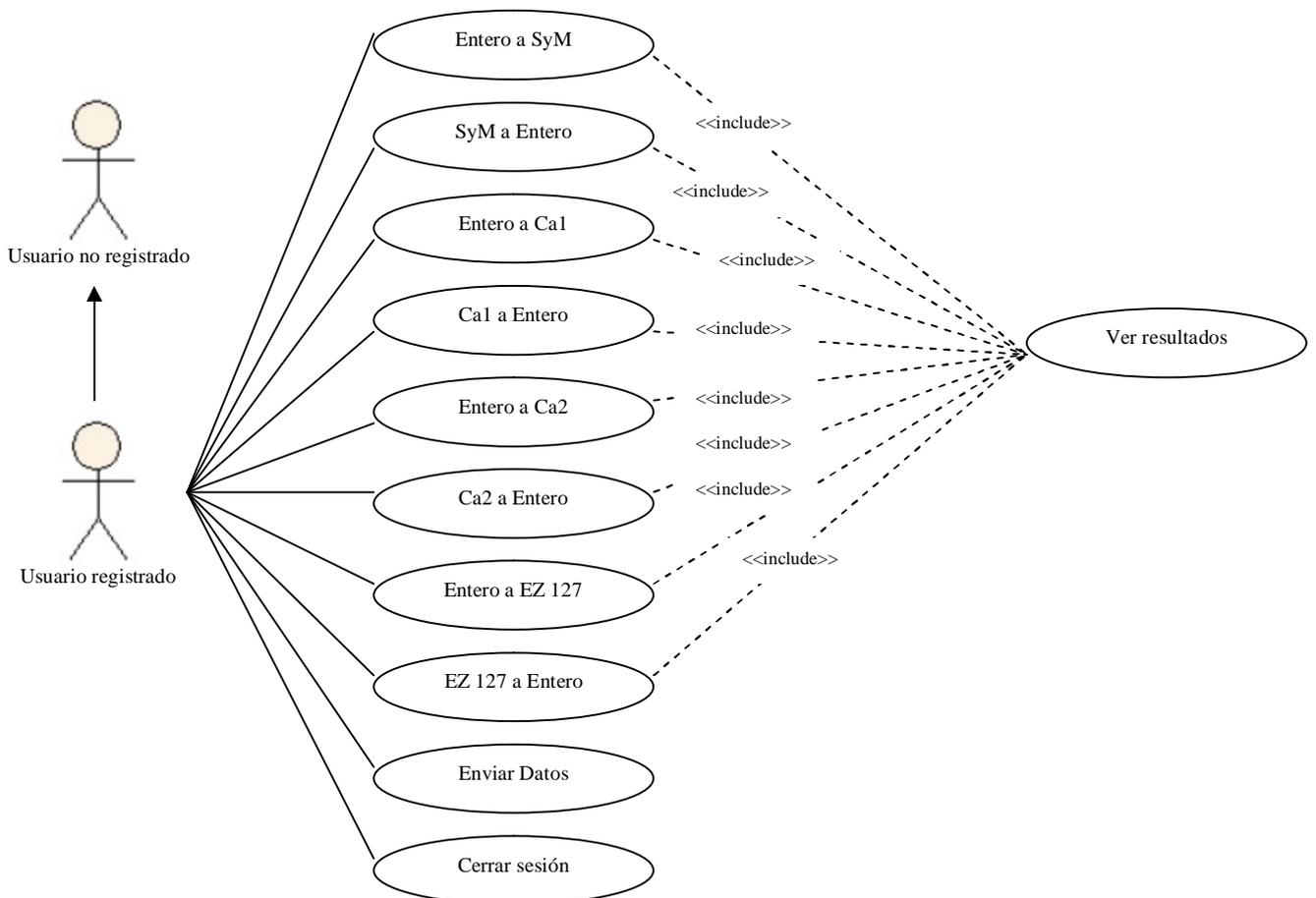
4 Análisis

En este apartado se describen la estructura y funcionalidad del sistema mediante un diagrama UML (Lenguaje Unificado Modelado), en nuestro caso el diagrama de Casos de Uso.

4.1 Casos de Uso

Un diagrama de Casos de Uso nos permite especificar el comportamiento de los diferentes personajes o entidades, denominados actores, con el sistema.

En nuestro caso, un usuario no registrado puede convertirse en usuario registrado, y tener acceso a las diferentes funcionalidades del sistema.



5 Diseño

El diseño se ha realizado mediante la programación por capas en una arquitectura cliente-servidor, para separar la lógica de negocio de la lógica de diseño.

Estaría separada en 3 capas:

- **Capa de presentación:** también llamada “capa de usuario”, sirve al usuario para interactuar con la aplicación. En esta capa, se le presenta el sistema, se le comunica la información y se recibe la participación del usuario con la aplicación. Tiene que cumplir dos principios, tiene que ser entendible y fácil de utilizar, lo que se conoce como amigable o “friendly”.
- **Capa de negocio:** donde se encuentran los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras la ejecución. Está en contacto directo con la capa de presentación, recibiendo las solicitudes y devolviendo los resultados, y con la capa de datos, para enviar o solicitar datos a la base de datos.
- **Capa de datos:** aquí encontramos los datos de la aplicación. Puede recibir datos para guardar, o peticiones para devolver datos guardados, siempre desde la capa de negocio.

5.1 Capa de presentación

La aplicación dispone de 3 interfaces principales.

1. **index.php:** Página del login, donde un usuario ya registrado con anterioridad puede entrar en su sesión rellenando el campo “Usuario” que correspondería a su DNI y el campo “Contraseña”. Al enviar la petición mediante el botón “Enviar consulta”, el formulario busca la pareja de datos formada por el Usuario y la Contraseña en la tabla “alumnos” de la base de datos “pfc”.
 - Si la encuentra, permite el registro y redirecciona el navegador a la página indexES.php, donde el alumno puede empezar a realizar los ejercicios.
 - En caso de no encontrarla, vuelve a solicitar la introducción de ambos campos (Usuario y Contraseña).

Si el usuario nunca se ha registrado en la aplicación, el alumno puede darse de alta pinchando en el enlace “Crea un nuevo usuario”. Este enlace redirecciona el navegador a la página creación.php, donde encontrará un formulario con los datos necesarios para el login.



Login PFC

Usuario: Contraseña:

[Crea un nuevo usuario](#)

2. **creacion.php:** Página para darse de alta en la aplicación y convertirse en un usuario registrado.

En esta sección, el alumno debe rellenar un formulario con sus datos personales y, mediante el botón “Crear Alumno”, enviar la solicitud de creación a la base de datos “pfc”.

El formulario realiza las siguientes comprobaciones antes de enviar los datos a la base de datos:

- Cheque la estructura del correo electrónico
- Chequea que el DNI es correcto (longitud y letra de control). Si es correcto, comprueba si existe una coincidencia en la tabla alumnos de la base de datos:
 - Si existe coincidencia en la base de datos, devuelve un aviso de que el DNI ya existe y redirecciona el navegador a creacion.php
 - Si no hay coincidencia, actualiza la tabla y redirecciona la página a login.php para que el alumno pueda realizar el login y acceder a la aplicación

Rellene el formulario para darse de alta

DNI

CONTRASEÑA

APELLIDOS

NOMBRE

CORREO ELECTRONICO

GRUPO

Información de navegador y sistema:
Mozilla/5.0 (Windows NT 6.1;
WOW64; rv:8.0) Gecko/20100101
Firefox/8.0

3. **indexES.php:** Página de ejercicios, donde el alumno realmente interactúa con la aplicación. Puede realizar las siguientes funciones:

- Seleccionar el tipo de ejercicio a realizar
- Ver el resumen de sus estadísticas
- Enviar los resultados al servidor.
- Cerrar sesión



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Proyecto Fin de Carrera - Jose Sánchez Baviera

Hola, Jose. Pulsa [AQUI](#) para cerrar sesion

Menú

[Entero a Signo y Magnitud](#)

[Signo y Magnitud a Entero](#)

[Entero a Complemento a 1](#)

[Complemento a 1 a Entero](#)

[Entero a Complemento a 2](#)

[Complemento a 2 a Entero](#)

[Entero en Exceso 127](#)

[Exceso 127 a Entero](#)

Ejercicios: 0	Aciertos: 0	Errores: 0	Tiempo: 4
Apellidos:	Sanchez Baviera	Nombre:	Jose
Mail:	josanba5@ei.upv.es	Grupo:	tardes
<input type="button" value="EnviarDatos"/>		<input type="button" value="EnviarDatosSQL"/>	

indexES.php

Una vez seleccionado el tipo de ejercicio, este es representado en la parte central o “contenido”.



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Proyecto Fin de Carrera - Jose Sánchez Baviera

Hola, Jose. Pulsa [AQUI](#) para cerrar sesion

Menú

- Entero a Signo y Magnitud
- Signo y Magnitud a Entero

Representar 37 en signo y magnitud con 8 bits

Signo: Magnitud:

En este momento, aparecen dos opciones:

- 1) “Comprobar”: Chequea si la respuesta es correcta.
 - Si es correcta nos permite seleccionar entre realizar otro ejercicio similar (“Continuar”) o salir de la sección (“Finalizar”) para seleccionar otro tipo de ejercicio o finalizar la sesión



- Si es incorrecta, proporciona una pista para que el alumno pueda solucionarlo, y le proporciona otra oportunidad para que realice el ejercicio

Representar 37 en signo y magnitud con 8 bits

Signo: Magnitud: Comprueba el signo

- 2) “Desistir”: El alumno abandona el ejercicio y vuelve al inicio para poder seleccionar otro tipo de ejercicio, enviar los resultados o finalizar la sesión.

5.2 Capa de negocio

Aquí estarían incluidos todos los ficheros JavaScript y php para el cálculo de los ejercicios, el chequeo de las respuestas y el envío de los resultados.

Dentro de la carpeta js, podemos encontrar:

- **proyecto.js:** Aquí se definen las variables para las funciones y las funciones necesarias para la presentación de los ejercicios, el chequeo de los mismos, y el envío de datos. Este fichero JavaScript es el principal, y se apoya en el resto de ficheros php.
- **proyecto.php:** Requiere el fichero Encriptacion.php. Genera el fichero txt con los resultados del alumno que se envía al servidor.
- **proyectoSQL.php:** Requiere el fichero Encriptacion.php. Envía los resultados del alumno a la tabla “ejercicios” de la base de datos “pfc” en MySQL.
- **encripta.js:** Contiene las funciones básicas en JavaScript para la encriptación de los datos.
- **Encriptacion.php:** Encripta los datos apoyándose en las funciones de encripta.js, para el envío al servidor.

Dentro de la carpeta php, encontramos:

- **login.php:** Permite al alumno registrarse en la aplicación y tener acceso a los ejercicios
- **logoff.php:** Cierra la sesión iniciada por un alumno

5.3 Capa de datos

Para el almacenamiento de datos, la aplicación se apoya en una base de datos MySQL denominada “pfc” gestionada por phpMyAdmin.

La estructura está incluida en el fichero llamado pfc.xml incluido en la carpeta db.



BDD: pfc

La bdd “pfc” contiene dos tablas:

1) **alumnos**, donde almacenamos los datos necesarios para el login:

- apellidos
- nombre
- correo
- grupo
- **dni**
- contraseña



Tabla “alumnos”

2) **ejercicios**, donde almacenamos los resultados del alumno para la evaluación del tutor:

- **dni**
- ejerciciosEN2SyM, aciertosEN2SyM, erroresEN2SyM
- ejerciciosSyM2EN, aciertosSyM2EN, erroresSyM2EN
- ejerciciosEN2C1, aciertosEN2C1, erroresEN2C1
- ejerciciosC12EN, aciertosC12EN, erroresC12EN
- ejerciciosEN2C2, aciertosEN2C2, erroresEN2C2
- ejerciciosC22EN, aciertosC22EN, erroresC22EN
- ejerciciosEN2EZ, aciertosEN2EZ, erroresEN2EZ
- ejerciciosEZ2EN, aciertosEZ2EN, erroresEZ2EN
- tiempo, fecha, firma

#	Columna	Tipo	Cofejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	1 dni	varchar(9)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	2 ejerciciosEN2SyM	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	3 aciertosEN2SyM	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	4 erroresEN2SyM	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	5 ejerciciosSyM2EN	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	6 aciertosSyM2EN	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	7 erroresSyM2EN	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	8 ejerciciosEN2C1	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	9 aciertosEN2C1	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	10 erroresEN2C1	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	11 ejerciciosC12EN	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	12 aciertosC12EN	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	13 erroresC12EN	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	14 ejerciciosEN2C2	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	15 aciertosEN2C2	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	16 erroresEN2C2	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	17 ejerciciosC22EN	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	18 aciertosC22EN	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	19 erroresC22EN	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	20 ejerciciosEN2EZ	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	21 aciertosEN2EZ	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	22 erroresEN2EZ	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	23 ejerciciosEZ2EN	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	24 aciertosEZ2EN	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	25 erroresEZ2EN	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	26 tiempo	varchar(255)	utf8_spanish_ci		No	Ninguna		
<input type="checkbox"/>	27 fecha	varchar(255)	utf8_spanish_ci		No	Ninguna		

Tabla “ejercicios”

6 Implementación

Esta parte la vamos a dividir en tres grandes bloques:

6.1 BLOQUE I: Login

En esta parte, el alumno necesita estar registrado para poder acceder al bloque 2. La página principal del proyecto, `index.php`, es un formulario que solicita el Usuario y la Contraseña para poder identificarse. Ambos campos son enviados a `login.php`, que compara ambos campos con la tabla “alumnos” de la base de datos. Si el DNI existe y la Contraseña es idéntica, realiza el login y redirecciona la página a `indexES.php`. Si el usuario no es válido o no existe, no nos permite acceder a los ejercicios y nos devuelve a `index.php`, para que introduzcamos unos valores válidos.

```
<?php
    if(isset($_SESSION['login'])){
        echo "Hola, ".$_SESSION['nombre'].". Pulsa <a href='../php/logoff.php'>AQUI</a> para cerrar sesion";
    }else{
        echo '<form id="formlogin" method="post" action="php/login.php">
        <tr><td>Usuario: </td><td><input type="text" name="usuariologin"></td></tr>
        <tr><td>Contraseña: </td><td><input type="password" name="contrasenalogin"><br></td></tr>
        <br>
        <tr><td><input type="submit"></td></tr>
        </form>
        <br>
        <a href="html/creacion.php">Crea un nuevo usuario</a><br>';
    }
?>
```

`index.php`

```
<?php
session_start();

$usuarioenviado=$_POST['usuariologin'];
$contrasenaenviada = $_POST['contrasenalogin'];

$conexion = mysql_connect("localhost","pfc_owner","pfc_owner");
mysql_select_db("pfc",$conexion);
$peticion = mysql_query("SELECT * FROM alumnos WHERE dni = '".$_.$usuarioenviado.'" AND contrasena = '".$_.$contrasenaenviada.'";");
if($fila = mysql_fetch_array($peticion)){
if(isset($fila['dni'])){
    $_SESSION['login'] = "true";
    $_SESSION['dni'] = $usuarioenviado;
    $_SESSION['nombre'] = $fila['nombre'];
    $_SESSION['apellidos'] = $fila['apellidos'];
    $_SESSION['correo'] = $fila['correo'];
    $_SESSION['grupo'] = $fila['grupo'];
}
mysql_close($conexion);
echo '<meta HTTP-EQUIV="REFRESH" content="0; url=../html/indexES.php">';
}else{mysql_close($conexion);
echo '<meta HTTP-EQUIV="REFRESH" content="0; url=../index.php">';
}
?>
```

`login.php`

Una vez iniciada la sesión, desde indexES.php, podemos finalizar la sesión pinchando en un enlace donde ejecutamos logoff.php

```
<?php
session_start();
session_destroy();

?>
<meta HTTP-EQUIV="REFRESH" content="0; url=../index.php">
```

logoff.php

Si es nuestra primera vez en la aplicación, podemos darnos de alta mediante un enlace que nos traslada a creación.php, donde podemos darnos de alta en la base de datos. Como comprobación, el usuario será el dni del alumno, el cual tiene que ser válido y único, no existir antes en la base de datos. La otra comprobación que realizamos antes de enviar los datos a la base de datos es que la estructura del correo electrónico sea la correcta. Al rellenar el formulario y enviarlo mediante el botón Crear Alumno, enviamos los datos a createuser.php, donde actualiza la tabla “alumnos”, siempre que el DNI no exista con anterioridad.

```
<?php
session_start();

$dni = $_GET['dni'];
$contrasena = $_GET['contrasena'];
$apellidos = $_GET['apellidos'];
$nombre = $_GET['nombre'];
$mail = $_GET['mail'];
$grupo = $_GET['grupo'];

$conexion = mysql_connect("localhost","pfc_owner","pfc_owner") or die(mysql_error());
mysql_select_db("pfc",$conexion);
$insertar = mysql_query("INSERT INTO alumnos (apellidos, nombre, correo, grupo, dni, contrasena)
VALUES ('{$apellidos}', '{$nombre}', '{$mail}', '{$grupo}', '{$dni}', '{$contrasena}'), $conexion);
if (!$insertar) {
echo "
<script language='JavaScript'>
    alert('Usuario no creado, DNI duplicado');
</script>";
echo '<meta HTTP-EQUIV="REFRESH" content="0; url=../html/creacion.php">';
die("Fallo en la insercion de registro en la Base de Datos: " . mysql_error());
}
mysql_close($conexion);
echo "
<script language='JavaScript'>
    alert('Creacion correcta, tienes que logearte');
</script>";

echo '<meta HTTP-EQUIV="REFRESH" content="0; url=../index.php">';
?>
```

createuser.php

6.2 BLOQUE II: Presentación de ejercicios

Una vez logeados, el alumno accede a la página de ejercicios, llamada indexES.php.

En esta página están incluidos todos los tipos de ejercicios (incluidos en “div” diferenciados) y dependiendo la opción del menú que selecciona el alumno, muestra un determinado ejercicio u otro.

La estructura es simple. Se divide en 4 frames.

- 1) En la parte superior o “encabezado”, nos muestra el título de la página, un saludo al alumno que ha iniciado la sesión y una opción para finalizar la misma, que redirecciona la página a index.php.
- 2) En la parte izquierda o “menu”, nos muestra el menú para seleccionar el tipo de ejercicio que el alumno quiere practicar.
- 3) En la parte central derecha o “contenido”, nos muestra los ejercicios para resolver, así como el formulario donde el alumno puede enviar la respuesta para su chequeo, o un botón donde desiste del ejercicio representado, y le permite seleccionar otro ejercicio diferente.
- 4) En la parte inferior o “pie”, hay una tabla con las estadísticas del alumno. Un contador del número de ejercicios realizado, el tiempo empleado y un pequeño resumen de los datos del alumno.

En el momento que el navegador carga la página indexES.php, ejecuta la función InicializarPagina, incluida en el fichero proyecto.js:

- InicializarPagina: Esta función nos muestra el “div” menu1 para que el alumno pueda escoger el tipo de ejercicio a practicar del menú, y deja no visibles el resto de “div” incluidos en el “contenido”, a la espera de que el alumno seleccione un tipo de ejercicio a resolver.

Una aclaración importante sobre la manera en que se ha implementado la página de ejercicios es la representación de los mismos. No se ha optado por crear una página html diferente por cada tipo de ejercicio, sino que están todos incluidos en la misma página html, en div diferenciados dentro del mismo div general (contenido).

Esto implica que cada vez que el alumno selecciona desde el menú un tipo de ejercicio, hay que ocultar todos los div incluidos en contenido, y mostrar el que incluye la pregunta junto con el formulario seleccionado.

6.2.1 Lógica de los ejercicios

Para la presentación y evaluación de los ejercicios, nos basamos en un fichero JavaScript denominado **proyecto.js**.

En este fichero es donde se incluyen la mayoría de funciones JavaScript para la representación de los ejercicios, el chequeo de los mismos, y el envío, apoyándose en funciones php, de los resultados obtenidos.

Primero definimos todas las variables que vamos a utilizar, tanto para el contador como para las funciones que resolverán los ejercicios.

- Variables de apoyo necesarias para el cálculo y representación de los ejercicios
 - aleat
 - signo
 - magnitud

- Variables contadores, donde guardamos el número de ejercicios realizados, los aciertos y los errores por cada tipo de ejercicio
 - ejerciciosTIPO
 - aciertosTIPO
 - erroresTIPO

- Variables de datos, donde guardamos los datos del alumno, el navegador y la “clave” para el encriptado de datos.
 - DNI
 - Contraseña
 - Apellidos
 - Nombre
 - Mail
 - Grupo
 - clave
 - navegador

- Variables de tiempo, para guardar la fecha en que realiza los ejercicios y el contador del tiempo utilizado
 - fecha
 - tiempo

La siguiente parte incluye el esquema de representación y chequeo de ejercicios. La explicación sirve para los diferentes tipos de ejercicios a resolver ya que la lógica es similar en todos los casos:

- **Función generadora del ejercicio:**

Genera un número aleatorio comprendido entre las dos variables que le proporcionamos al ejecutar la función.

Apoyándose en una función específica de cada tipo de ejercicio, calcula el resultado de realizar la conversión y la almacena en una variable.

Imprime la pregunta en el “contenido” junto con el formulario para que el alumno escriba la solución.

En este momento, el alumno tiene dos opciones:

- 1) Pinchando en el botón “Desistir”, aumenta una unidad el contador de errores del tipo de ejercicio que había seleccionado, y vuelve al estado inicial de la página `indexES.php`
- 2) Pinchando en el botón “Comprobar” del formulario, ejecutamos la función `Check`.

- **Función `Check`:**

Compara el resultado enviado por el alumno a través del formulario con la solución almacenada en la variable, en el paso anterior.

- 1) Si es correcta, aumenta el contador de aciertos del tipo específico del ejercicio, y proporciona al alumno la opción de realizar otro ejercicio del mismo tipo, o de volver al estado inicial de la página, en la que podría seleccionar otro tipo de ejercicio, enviar los datos con los resultados o simplemente salir de su sesión.
- 2) Si es incorrecta, el formulario informa de que la respuesta es errónea proporcionando una pista para la resolución del sistema:
 - Comprueba el signo
 - Comprueba la magnitud
 - Magnitud debe tener x bits
 - Número incorrecto
 - Posibles combinaciones entre las pistas arriba indicadas

6.3 BLOQUE III: Envío de resultados

Una vez finalizada la ejecución de los ejercicios, el alumno puede optar por enviar los resultados al tutor para el seguimiento de su evaluación. El envío puede ser realizado de dos formas diferentes:

- Vía archivo txt
- A una tabla llamada “ejercicios”, dentro de la base de datos MySQL denominada “pfc”

En ambos casos, la información sufrirá una pequeña encriptación antes de ser enviada al servidor, y una vez allí será descriptada y comprobado que no ha sido modificada ni alterada en el envío.

Como la función de encriptar es similar independientemente de la forma de envío, vamos a explicarla tanto para el envío a un fichero txt en el servidor mediante la función `EnviarDatos()`, como para el envío a la base de datos mediante la función `EnviarDatosSQL()`.

- 1) Guardamos en la variable “firma”, la cadena resultante de los datos del alumno (Apellidos+Nombre+Mail+Grupo). Esta cadena es la que chequearemos para comprobar que no ha habido manipulación de los datos.
- 2) Guardamos en la variable “params”, todas las variables del alumno, tanto los datos personales, como los resultados de los ejercicios (realizados, acertados y fallados). Añadimos también las variables de tiempo empleado, la fecha y la firma.

Todas las variables, firma incluida, van a ser encriptadas por separado y enviadas al fichero php que las descriptará, a excepción de la firma que nos servirá para comprobar la veracidad de los datos recibidos, y enviará los datos al servidor.

6.3.1 Encriptación

Cada variable va a tener el mismo proceso de encriptación. Desde la función `EnviarDatos` o `EnviarDatosSQL`, incluidas en el fichero `proyecto.js`, vamos a realizar la encriptación de cada variable apoyándonos en el fichero `encripta.js`:

- 1) La función `Encripta` recibe el valor de la variable y una clave, en nuestro caso “pfc”, con el que calcularemos el desplazamiento.
- 2) La función `GetDesplazamiento`, envía cada carácter de la clave a la función `ConvertorANumero` para obtener un número que devolverá a la función `Encripta`.
- 3) Este número representa el desplazamiento, y desde `Encripta`, enviamos carácter a carácter de cada variable junto con el desplazamiento a la función `EncriptaLetra`. Esta función nos devuelve el carácter equivalente según el desplazamiento calculado, `letra_deplazada`, enviando cada carácter a la función `ConvertorALetra`.
- 4) Cada variable es encriptada por separado y enviada al fichero php correspondiente (`proyecto.php` o `proyectoSQL.php`) para el envío al servidor.

6.3.2 Desenscriptación

Una vez enviadas las variables firma y parámetros al servidor, correctamente encriptadas, el fichero php (proyecto.php o proyectoSQL.php) se encarga de la desenscriptación apoyándose en el fichero Encrptacion.php. El funcionamiento es similar al de encriptación pero a la inversa.

- 1) Cada variable es enviada a la función Desenscripta. Esta función envía la variable encriptada a la función Decripta.
- 2) Decripta utiliza la variable global clave, en nuestro caso “pfc”, para obtener el desplazamiento con la función GetDesplazamiento.
- 3) GetDesplazamiento envía carácter a carácter de la clave a la función ConversoANumeros, para obtener el desplazamiento utilizado y lo devuelve a Decripta.
- 4) Una vez calculado el desplazamiento utilizado, que debe coincidir con el desplazamiento calculado en la parte de encriptación, Decripta envía carácter a carácter de cada variable a la función DesenscriptaLetra, la cual devuelve el carácter original.
- 5) Una vez obtenida la variable desenscriptada, Desenscripta devuelve la cadena resultante al fichero php (proyecto.php o proyectoSQL.php).
- 6) Para confirmar que los datos no han sido vulnerados, volvemos a encriptar la firma (apellidos+nombre+mail+grupo), y la comparamos con la firma encriptada enviada desde la función JavaScript (EnviarDatos o EnviarDatosSQL).

7 Evaluación

En esta sección presentamos una serie de pruebas de validación sobre las diferentes partes del proyecto

7.1 Prueba de validación CSS

Para validar la hoja de estilos, se ha chequeado en la página oficial de W3C, en la sección de CSS Validation Service

Resultados del Validador CSS del W3C para proyecto.css (CSS versión 3)

¡Enhorabuena! No error encontrado.

¡Este documento es [CSS versión 3](#) válido!

Puede mostrar este icono en cualquier página que valide para que los usuarios vean que se ha preocupado por crear una página Web interoperable. A continuación se encuentra el XHTML que puede usar para añadir el icono a su página Web.

```
<p>
<a href="http://jigsaw.w3.org/css-validator/check/referer">

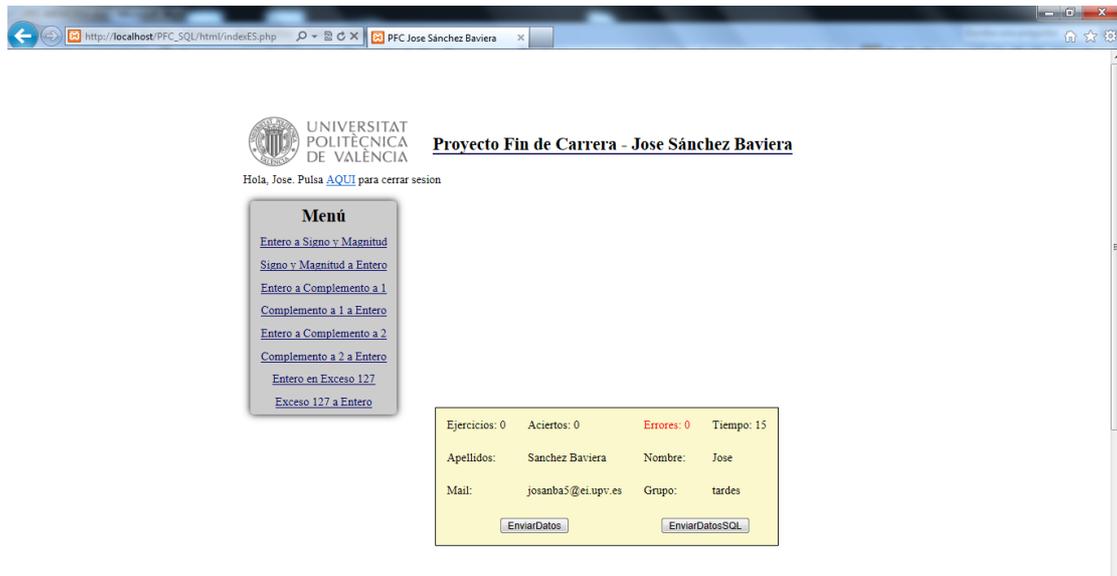
</a>
</p>
```

```
<p>
<a href="http://jigsaw.w3.org/css-validator/check/referer">

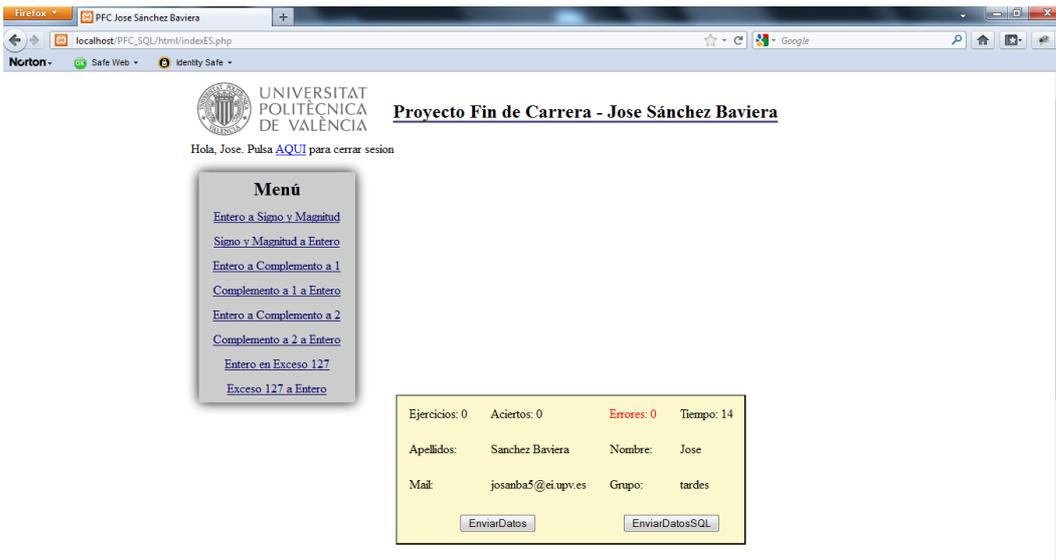
</a>
</p>
```

7.2 Prueba de validación sobre diferentes navegadores

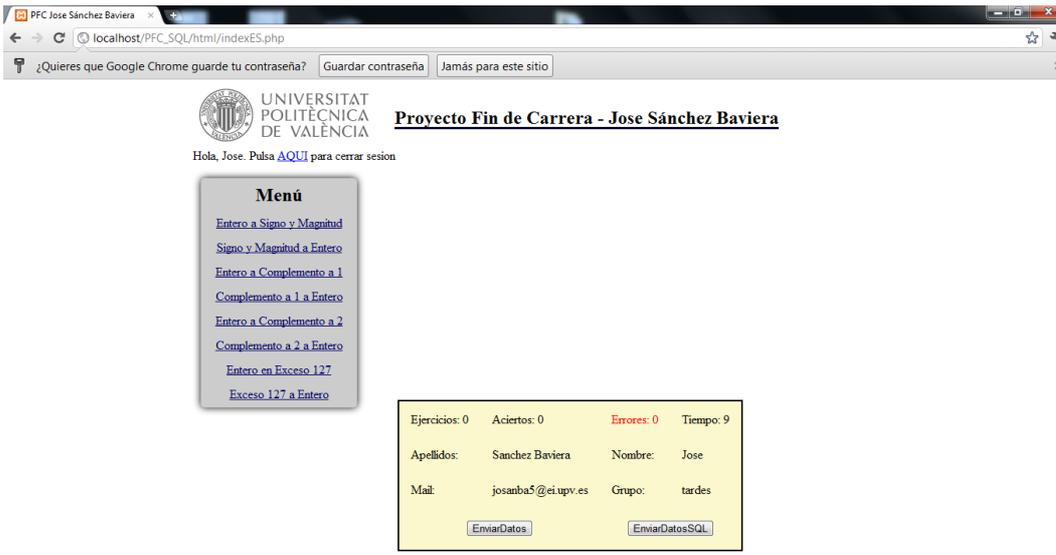
Para validar la correcta visión de la aplicación, ha sido chequeada la aplicación en los navegadores más comunes utilizados por los usuarios.



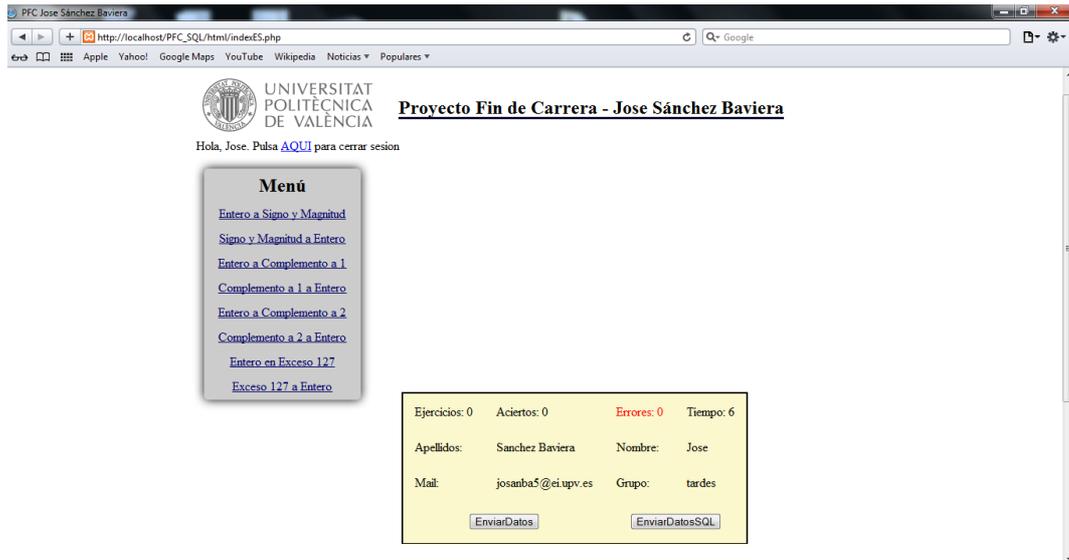
Internet Explorer



Mozilla Firefox



Google Chrome



Safari

8 Conclusiones

8.1 Valoración personal

El diseño y la funcionalidad de la aplicación cumplen los requisitos de la especificación del proyecto. Permite al alumno tener acceso a una Web donde poder practicar y entender mejor los ejercicios de representación de números enteros. Creo que puede ser útil tanto para el alumno, el cual puede obtener una herramienta adicional de estudio, como para el tutor, permitiéndole llevar un control y poder realizar una evaluación sobre los avances del alumno.

A nivel personal, he aprendido mucho sobre lenguaje JavaScript y Php, donde al principio del proyecto carecía de conocimientos suficientes. Debido a este déficit, la presentación del proyecto se ha demorado en el tiempo más del que pensaba sería necesario. Con apoyo de mi tutor, de compañeros y buscando información, tanto en la Web como realizando un curso sobre Php, he logrado finalizar la aplicación con un resultado bastante aceptable.

En conclusión, estoy satisfecho con el resultado de la aplicación, admitiendo que estoy seguro se podría haber realizado de una manera más completa, y tal vez más simple.

8.2 Posibles extensiones del proyecto

Después de haber realizado varias pruebas actuando como alumno, se confirma que la aplicación ofrece un número alto y suficiente de ejercicios diferentes para poder trabajar y mejorar en el entendimiento de la representación de número enteros.

Una posible extensión sería mejorar el chequeo de las respuestas incorrectas, y proporcionar al alumno una más amplia explicación sobre el fallo realizado y una guía más eficaz para la solución del ejercicio.

9 Bibliografía

- IEEE-STD-830-1998: Especificaciones de los requisitos del Software
- Página oficial de Xampp: <http://www.apachefriends.org/es/xampp.html>
- Apuntes Estructura de Computadores, EI

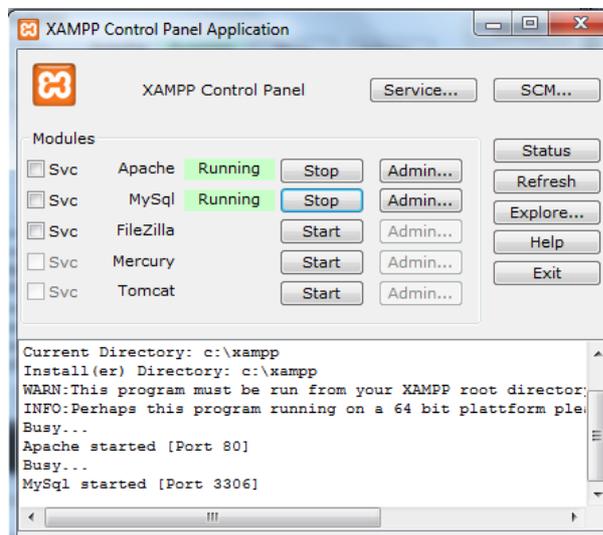
10 Anexos

10.1 Preparación entorno de trabajo

Primero de todo, vamos a preparar el entorno de trabajo para poder probar la aplicación. Descargamos XAMPP de la página oficial (<http://www.apachefriends.org/es/xampp.html>).

XAMPP es un servidor de software libre, que se puede utilizar en cualquier plataforma. Consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.

Una vez hemos realizado la instalación, procedemos a ejecutarlo. Nos hará falta ejecutar el servidor Apache y la base de datos MySQL.



Para comprobar que la instalación ha sido correcta, desde el navegador que utilizemos, entramos en localhost y debería de aparecer la página de inicio de XAMPP.

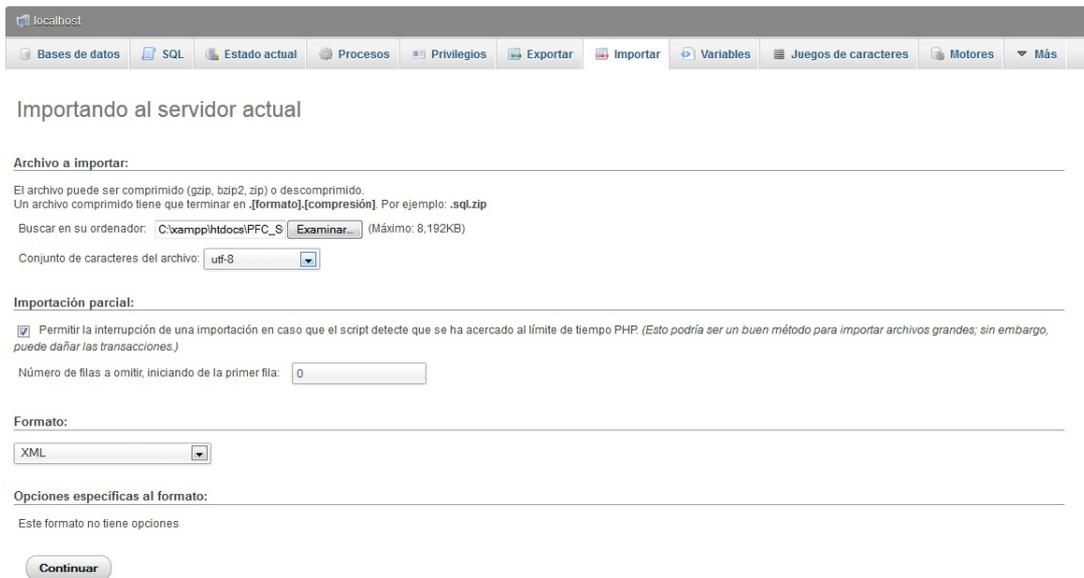


10.2 Configuración base de datos

Copiamos la carpeta PFC_SQL dentro de la siguiente ruta: c:/xampp/htdocs

Una vez dentro de la página principal de XAMPP, vamos a proceder a la importación de la base de datos “pfc”. El fichero pfc.xml se encuentra dentro de PFC_SQL/bd

Desde la herramienta phpMyAdmin, importamos la base de datos en formato XML tanto la estructura (tabla alumnos y tabla ejercicios), como el contenido de estas.



Una vez importada, aparece un usuario creado para el testeo de la aplicación:

Dni: 12345678Z

Contraseña: tester



Solo nos quedaría realizar un último paso antes de poder probar la aplicación.
 Dentro de la opción “Privilegios”, tenemos que crear un usuario para poder acceder a la base de datos desde la aplicación

Nombre: pfc_owner

Servidor: localhost

Contraseña: pfc_owner

Ahora ya podemos probar la aplicación, entrando con el usuario Tester