

## Práctica 1ª Evaluación DWES




1.	Versión del documento .....	2
2.	Normas de obligado cumplimiento .....	2
3.	Comentarios a la práctica.....	2
4.	Gestión Gimnasio .....	4
4.1	Requisitos funcionales .....	4
4.1.1	Gestión de usuarios o socios.....	4
4.1.2	Inicio de sesión .....	5
4.1.3	Discriminación del tipo de usuario .....	5
4.1.4	Gestión de actividades dirigidas.....	5
4.1.5	Gestión del Horario de las actividades dirigidas .....	6
4.1.6	Realización de las actividades dirigidas .....	6
4.1.7	Gestión de datos principales del gimnasio.....	7
4.1.8	Envío de mensajes de correo electrónico .....	7
4.1.9	Envío de mensajes internos a la aplicación.....	7
4.1.10	Generación de pdfs.....	7
4.1.11	Otros requisitos funcionales .....	8
4.2	Información que maneja nuestra aplicación .....	9
5.	Instalador de la aplicación.....	10
6.	Consideraciones finales .....	11
7.	Valoración de la práctica .....	11
7.1	Criterios de evaluación .....	13
8.	Instrucciones para entregar la práctica.....	13

# 1. Versión del documento

**Versión 1.** Borrador de la práctica primera evaluación DWES curso 2020-21. (04-11-2020)







# 2. Normas de obligado cumplimiento








Para la realización de la aplicación será obligatorio el uso del patrón **MVC** en el desarrollo de los problemas. Su no utilización se valorará con -3 puntos en la evaluación de la práctica.

-  El acceso a la base de datos se realizará mediante una capa de abstracción de datos creada por vosotros, utilizando un enfoque de POO con PDO.
-  El código de las distintas páginas deberá ser documentado con comentarios, indicando en cada fichero php el autor, la fecha de creación, la versión del fichero.
-  Se documentará cada una de las funciones y variables de clase que se creen utilizando el formato de algún programa de documentación automática (phpDocumentor o ApiGen): Básicamente lo que debéis hacer es abrir comentarios de tipo bloque `/** */` delante de cada función y variable de clase, y el entorno se encargará de generarlos automáticamente las etiquetas (@) pertinentes.

# 3. Comentarios a la práctica

La estructura de carpetas del proyecto y sus contenidos será la siguiente (se admiten variantes):

Tipo	Elemento	Descripción
 app		Carpeta general que contiene todos los ficheros de la aplicación. Controladores, vistas y modelos cada uno organizado en sus respectivas carpetas
	 index.php	Fichero que iniciará la aplicación
	 config.php	En este fichero se incluirán parámetros de configuración de la aplicación, como: <ul style="list-style-type: none"> <li>• Usuario y clave para acceder a la base de datos</li> <li>• Nombre de la base de datos</li> </ul> Cualquier otra información que consideréis relevante como: <ul style="list-style-type: none"> <li>• Ruta url de la aplicación</li> <li>• Ruta absoluta dentro del servidor</li> </ul> Todo lo anterior se almacenará en variables PHP que luego se utilizarán en la aplicación.
	 controllers  views  models	Carpetas en las que almacenaremos las distintas partes que componen la aplicación

	 install	En esta carpeta se incluirá cualquier información que se considere relevante para instalar la aplicación.
	 bd.sql	Fichero que contiene un script sql que crea la base de datos que utilizará el programa. En el script se creará <ul style="list-style-type: none"> <li>• La base de datos</li> <li>• La estructura de las tablas</li> <li>• Los usuarios si utilizase alguno diferente</li> </ul>
	 Assets	Imágenes, ficheros de estilos (css), ficheros de script (js), y demás ficheros utilizados en el diseño y desarrollo de la aplicación.
	 css	
	 img	
	 js	
 Doc		Documentación de la aplicación

- ⇒ Las distintas tareas y funcionalidades que se plantean en el proyecto tienen como objetivo el de obligaros a trabajar los contenidos vistos en clase, así como que profundicéis en vuestros conocimientos sobre la metodología de la programación (cómo resolver los problemas).
- ⇒ Esta práctica está pensada para que la realicéis compaginando el trabajo en clase y el trabajo en vuestra casa. Ante cualquier duda acerca de cómo afrontar un problema determinado deberíais buscar posibles soluciones en internet o, en última instancia, preguntar al profesor, pues se trata de fomentar el aprendizaje autónomo del alumnado.
- ⇒ El proyecto se expondrá al profesor para poder verificar su funcionamiento y autoría.

## 4. Gestión Gimnasio

Se pretende construir una aplicación web sobre una plataforma LAMP que gestione algunos aspectos básicos del funcionamiento de un gimnasio tipo como el que nos podemos encontrar en nuestra ciudad.

La gestión comprenderá, al menos, el sistema de usuarios/socios y el sistema de actividades dirigidas.

### 4.1 Requisitos funcionales

Los principales requisitos funcionales que debe cumplir nuestra aplicación son:

#### 4.1.1 Gestión de usuarios o socios

En nuestra aplicación existirán **dos roles principales**: el *administrador* y el *usuario normal* o *socio*.

- Los **administradores** son los usuarios encargados de **gestionar la aplicación**, teniendo **acceso a toda su información y funcionalidad**.
- Los **socios** son los usuarios que, tras registrarse en la web, pueden gestionar **su relación y uso del gimnasio**.

El **administrador** tendrá la capacidad de **consultar, dar de alta, dar de baja y modificar los datos de los usuarios o socios del gimnasio**.

Como **posibilidad extra** existirá la posibilidad de **dar de alta un conjunto de usuarios mediante la importación de sus datos desde un fichero tipo Excel**. Una vez hayan sido creados los **socios** estos **accederán a su cuenta con login y contraseña introduciendo su NIF en ambos casos**.

Si el **administrador así lo decide**, el sistema obligará a **cambiar la contraseña al entrar el socio por primera vez**.

**Cada socio podrá registrarse en la web mediante un formulario de registro**. El registro lo **validará el administrador** previa comprobación de los datos introducidos.

**Cada usuario** de la aplicación tendrá la posibilidad de **acceder y modificar los datos incluidos en su perfil (login, clave, nombre, apellidos, foto de perfil, etc)**.

#### 4.1.2 Inicio de sesión

La aplicación sólo deberá permitir acceder a su funcionalidad a los usuarios que previamente se hayan autenticado mediante un login y una clave.

En caso de intentar acceder directamente a alguna de las funciones, sin haberse validado previamente, se mostrará la pantalla de acceso al portal.

En el encabezado mostraréis en todo momento en la esquina superior derecha el nombre de usuario y la hora a la que ha iniciado la sesión, así como un enlace (texto, imagen o ambos) que permita finalizar la sesión.

Se dará la opción al usuario de mantener abierta la sesión así como el recuerdo de sus credenciales en la pantalla de login, de recuperar la contraseña y loguearse mediante las credenciales de alguna de las redes sociales (esto último es un requisito extra).

#### 4.1.3 Discriminación del tipo de usuario

Es obligatorio diferenciar el tipo de usuario que se ha validado, restringiendo las operaciones que pueda realizar en función de su perfil. Es decir, los socios sólo podrán realizar las operaciones que tienen asignadas, y tendrán ocultos los enlaces a las operaciones que están asignadas a los administradores.

En el encabezado de la aplicación se mostrará con claridad si el usuario es de tipo Socio o Administrador utilizando iconos.

#### 4.1.4 Gestión de actividades dirigidas

El usuario administrador podrá gestionar el alta, baja y modificación de la lista de actividades dirigidas ofrecidas por el gimnasio.

Son ejemplos de actividades dirigidas el *bodybalance*, *bodypump*, *yoga*, *táбата gt*, etc.

Vamos a suponer que tenemos en nuestro gimnasio una sala de usos múltiples con disponibilidad de sitio para las actividades dirigidas, de manera sólo puede desarrollarse una actividad simultánea en el gimnasio. En principio no se tendrá en cuenta la existencia de múltiples salas en el gimnasio.

Cada actividad dirigida contará con un aforo de participantes limitado y se podrá realizar en un determinado tramo horario dentro de las instalaciones del gimnasio. Este tramo horario vendrá determinado por el horario de apertura del gimnasio.

#### 4.1.5 Gestión del Horario de las actividades dirigidas

El administrador del gimnasio establecerá un horario semanal para la realización de las actividades dirigidas. Este horario estará comprendido dentro del horario de apertura del gimnasio: de 7 AM hasta las 11 PM de lunes a sábado (configurable).

El administrador seleccionará la actividad dirigida y le asignará las sesiones semanales posibles para su realización. Las sesiones tendrán una duración múltiplo de 15 minutos con un mínimo de 30 minutos y un máximo de 1 hora y media; además empezarán en una hora en punto, en una mitad de hora o en un cuarto de hora.

En función del horario semanal, los socios podrán reservar su participación en las actividades.

El horario semanal puede ser modificado cada vez que sea necesario, pero los cambios sólo se pueden aplicar a días futuros, es decir, no es posible cambiar el horario de sesiones que ya se han impartido o que se están impartiendo en el mismo día. Tampoco es posible cambiar fechas pasadas.

Como hemos supuesto que el gimnasio sólo tiene una sala de usos múltiples, un posible ejemplo de horario sería el siguiente:

Hora	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
7:00 – 7:30				Táбата		Táбата
7:30 – 8:00	AquaExpress		AquaExpress		A.I. Core	
7:45 – 8:30		Ciclo		Ciclo		Ciclo
8:15 – 8:45	Yoga		Fitball		CXWORX	
8:45 – 9:45	BodyPump	Ciclo	Gimnasia suave		BodyCombat	BodyBalance
9:45 – 10:15	A.I. Core			Yoga		
10:00 – 11:00		Zumba	Ciclo		Zumba	Pilates
...	...	...	...	...	...	...

No debería ser posible utilizar la sala al mismo tiempo en más de una actividad. Se tendrá en cuenta en la valoración de la práctica la complejidad del sistema de gestión de horario programado.

#### 4.1.6 Realización de las actividades dirigidas

Los socios que estén interesados en apuntarse a las actividades dirigidas podrán hacerlo desde la web del gimnasio.

Para eso entrarán en el portal del gimnasio y verán el horario semanal de actividades. Podrán seleccionar un máximo de dos actividades diarias. Es posible que la actividad tenga el aforo completo por lo que esa actividad no será elegible por el socio. Los socios pueden reservar actividades dentro de la semana en curso y la siguiente como máximo.

El sistema llevará el historial de las actividades que realizan los socios.

Los socios también podrán borrarse de las actividades siempre y cuando estas todavía no hayan tenido lugar.

#### 4.1.7 Gestión de datos principales del gimnasio

El administrador podrá configurar alguna información general de su gimnasio:

- ⇒ El nombre del mismo
- ⇒ El horario de apertura. Un rango de apertura dentro del día.
- ⇒ Días a la semana que abre el gimnasio.
- ⇒ Días festivos o de no apertura del gimnasio.
- ⇒ Otros que resulten interesantes.

#### 4.1.8 Envío de mensajes de correo electrónico

Esta opción se encuentra disponible sólo para administradores. Los administradores podrán enviar mensajes individuales a los socios que tengan algún tipo de incidencia, a todos los socios seleccionados de una lista mediante el uso de casillas de verificación, o bien, si dejamos el campo en blanco, se enviaría el mensaje a todos los usuarios registrados.

#### 4.1.9 Envío de mensajes internos a la aplicación

Todos los usuarios podrán enviar mensajes dirigidos a otros usuarios de la aplicación.

Desde la pantalla de inicio se deberá implementar una especie de bandeja de entrada/salida de mensajes que permita gestionarlos. Desde ahí se podrá enviar un mensaje de texto a un usuario y recibir los mensajes de otros usuarios.

#### 4.1.10 Generación de pdfs

Esta opción estará disponible para todos los usuarios registrados. Los administradores podrán generar PDFs de todos los socios de cualquiera de los listados que generemos, de las actividades dirigidas, del horario y de los usuarios que hacen las actividades. También podremos generar el PDF de un socio en concreto yendo a la lista de socios o en el perfil de dicho socio. Siendo socio, podremos generar el PDF de nuestro perfil y de las reservas de actividades.

#### 4.1.11 Otros requisitos funcionales

- Utilizar el framework CSS **Bootstrap** así como **Google fonts** para obtener un diseño más atractivo y **responsive**.
- **Organizar y dividir las distintas secciones** que componen las páginas de la aplicación **en distintos ficheros** a fin de aumentar la modularidad y facilitar el mantenimiento y la escalabilidad, **de forma que en todo momento estemos viendo un encabezado, menú y pie común a toda la aplicación.**
- Se deben **filtrar, sanear y validar todos los datos antes de enviarlos al servidor, mostrando el pertinente error en el formulario, junto al campo correspondiente y manteniendo los valores enviados para que se puedan editar.**
- **Las operaciones que se realicen sobre la base de datos estarán verificadas mediante el uso de transacciones.**
- Es importante que en todo momento el **usuario sepa en qué parte de la aplicación se encuentra**, y tenga **opción de realizar las opciones principales de manera directa.**
- La aplicación mostrará **una página inicial**, desde la que se podrán realizar **todas las operaciones antes mencionadas a través de una barra de herramientas o por medio de una serie de botones asociados a las principales tareas implementadas.**
- Respecto al control de actividad de la aplicación, se va a disponer de **una tabla (log)** que **registrará los accesos y las operaciones que los usuarios realizan sobre la aplicación.** La **información** que almacenará dicha tabla **sería** la siguiente:
  - **Usuario.** Usuario que interactúa con la aplicación.
  - **Perfil.** Tipo de usuario que realiza la operación.
  - **Fecha y Hora de actividad.** Instante en el que el usuario realiza la operación.
  - **Tipo de actividad.** Indica qué operación realizó el usuario en la aplicación. Existen cuatro tipos de operaciones: acceso, alta, baja y actualización.
- La **gestión de esta tabla se realizará obligatoriamente mediante el uso de llamadas a procedimientos almacenados en MySQL.**



## 4.2 Información que maneja nuestra aplicación

De forma orientativa se incluye una posible estructura de las tablas que maneja nuestra aplicación. El alumno puede hacer modificaciones que respeten el enunciado de la práctica.

### Para los **usuarios**:

id	entero, clave de la tabla
nif	cadena
nombre	cadena
apellido1	cadena
apellido2	cadena
imagen	cadena, trayecto a la imagen de perfil del usuario
login	cadena, único
password	cadena, se guardará cifrada
email	cadena
telefono	entero
direccion	cadena
rol_id	int, hace referencia al rol del usuario

### Para **rol** de usuarios

id	entero, clave de la tabla
tipo	cadena, puede ser admin o socio

### Para las **actividades** dirigidas:

id	entero, clave de la tabla
nombre	cadena
descripcion	cadena
aforo	entero

### Para los **tramos** del horario del gimnasio:

id	entero, clave de la tabla
dia	cadena, día de la semana del tramo: Lunes, Martes,...
hora_inicio	fecha, hora de comienzo del tramo horario
hora_fin	fecha, hora de fin del tramo horario
actividad_id	entero, referencia a actividad realizada en el tramo
fecha_alta	fecha, fecha en la que se definió el tramo
fecha_baja	fecha, fecha en la se dio de baja el tramo, null si sigue vigente

Los campos *fecha\_alta* y *fecha\_baja* posibilitan una forma de guardar un histórico del horario de actividades que ha tenido el gimnasio.

Para **tramo\_usuario**, relación n:m entre los usuarios (socios) y los tramos del horario. Vendría a representar la reserva de las actividades.

id	entero, clave de la tabla
tramo_id	entero, referencia al tramo horario
usuario_id	entero, referencia al usuario que reserva
fecha_actividad	fecha, fecha en la que se hará la actividad
fecha_reserva	fecha, fecha en la que se hace la reserva

La *fecha de la actividad* es la **fecha real en la que se hará la actividad**. Hay que **tener en cuenta** que el tramo horario no especifica una fecha concreta sino sólo el día de la semana.

La *fecha de la reserva* es la fecha en la que **el usuario se apunta a la actividad**. Los usuarios sólo pueden apuntarse a actividades que se realicen en el futuro, no en el pasado o en el día en curso.

Para los **mensajes**

id	entero, clave de la tabla
usu_origen	entero, id del usuario que manda el mensaje
usu_destino	entero, id del usuario destinatario del mensaje
mensaje	cadena

## 5. Instalador de la aplicación

Un instalador en una aplicación Web es una **aplicación que permite crear/configurar los parámetros de una aplicación web**. Los parámetros configurables son:

- **Datos para acceder a la base de datos (usuario, clave) y nombre de esquema**. Dichos datos los **almacenará en un fichero de configuración**. En vuestro caso esto no será preciso pues ya lo habréis creado con el editor en el fichero "app/config.php"
- **Creación de la estructura de la base de datos**. El instalador **creará la estructura de las tablas**, **definirá los procedimientos almacenados e inicializará los datos que se precisen**. Puede que incluso llegue a crear la base de datos, aunque esto es menos habitual. Si es preciso, el instalador modificará otros parámetros funcionales que necesitare la aplicación.

Se precisa realizar un instalador para la aplicación. El instalador estará situado en la carpeta "install", y arrancará automáticamente "index.php". Cogera los datos de configuración del fichero "app/config.php" y procederá a la creación e inicialización de la base de datos.

En el fichero **config.php** tendremos que poder definir:

- Ubicación del servidor de base de datos.
- Usuario que accede a la base de datos.
- Clave del usuario que accede a la base de datos.
- Base de datos con la que se trabajará.

El instalador borrará todas las tablas que existan en la base de datos y luego creará la estructura de las tablas con las que trabajará. Para ello puede que preciséis obtener la lista de tablas existentes en la base de datos para luego ir borrando cada una.

## 6. Consideraciones finales

- Se valorará positivamente el uso y personalización de una plantilla Bootstrap para la estructura de nuestro proyecto.
- En el desarrollo web, cuando diseñamos una tabla para una base de datos siempre es conveniente identificar los registros mediante un campo "id" de tipo numérico que se generará automáticamente, el cual será la clave principal. Esto facilitará más adelante la codificación de la funcionalidad de la aplicación.
- Debéis generar la documentación de la aplicación realizada, mediante el uso de phpDocumentor o Apigen, incluyendo los comentarios pertinentes y generando, finalmente, los documentos de forma automatizada.
- Es obligatorio que el alumno vaya publicando las diferentes revisiones de su código utilizando GitHub en algún repositorio público (tipo GIT) al que pueda acceder el profesor como medio de verificación de su autoría.
- En las webs 123RF, flaticon, freepik, iconmonstr, Gnome-look, iconos8, etc..., podéis encontrar múltiples paquetes de iconos gratuitos que os permitirán realizar un diseño de aplicación atractivo y coherente.
- En los sitios Template Monster y Theme Forest podréis encontrar numerosas plantillas web, algunas gratuitas, que os pueden servir de inspiración para el diseño de la aplicación.

## 7. Valoración de la práctica

La valoración de la práctica, desglosada por funcionalidades implementadas y funcionales sería la siguiente:

0.5 puntos – Instalador de la aplicación.

0.5 puntos – Control de acceso a la aplicación. (Login, Registro, etc)

0.25 puntos – Permitir recordar y mantener sesión abierta.

0.25 puntos – Activación de usuarios por parte del administrador.

0.5 puntos – Opción de Olvidó su contraseña.

0.5 puntos – Añadir Captcha en la operación de registro de usuario.

0.75 puntos – Validarse con las credenciales de Google o Facebook.

1 punto – CRUD de socios/usuarios.

1 punto – CRUD de actividades dirigidas.

0.5 puntos – Discriminación de tareas por perfiles de usuarios.

1.25 puntos – Gestión de horario de actividades del gimnasio. CRUD.

0.5 puntos – Visualización horario gimnasio

0.75 puntos – Gestión de reservas de actividades por los socios

1 punto – Listados de elementos de la aplicación: usuarios, actividades, reservas, mensajes...

1 punto – Buscar o filtrar en los listados utilizando distintos campos.

1 punto – Ordenar los resultados de listados ascendentemente/descendentemente por columnas.

0.75 puntos – Paginación en los listados.

0.75 puntos – Generar listados en pdf.

1 punto – Gestionar el envío y recepción de mensajes entre usuarios

0.75 puntos – Gestionar el envío de correos electrónicos entre usuarios.

0.75 puntos – Añadir bloque usuarios con página Excel .

1 punto – Mantener actualizada y listar el contenido de la tabla de control de actividad (Log) mediante procedimientos almacenados.

1 punto – Calidad del diseño de la interfaz de la aplicación/Uso de plantilla Bootstrap/Jquery/Ajax,....

0.25 puntos – Control de versiones.

0.75 puntos – Documentación (PhpDocumentor o ApiGen) de la aplicación y manual de usuario.

0.25 puntos – Desplegar la aplicación sobre Docker o Vagrant. (RA1)

### 0.5 puntos – Desplegar la aplicación sobre Heroku. (RA1)

#### -3 puntos – No utilización del patrón MVC

La nota de la práctica supondrá el 35% de la calificación sobre el bloque de contenido evaluado, por lo que su entrega será obligatoria.

La copia de todo o parte de la práctica supondrá la inmediata eliminación de la parte copiada. Se dividirá la nota de los implicados entre el número de copias. Igualmente, en caso de identificar a la persona que ha copiado, se le penalizará en su calificación.

Cada alumno mostrará y explicará, de manera individual el funcionamiento de la aplicación al profesor en la fecha que se acuerde tras su entrega.

La puntuación de cada ejercicio es la que se indica en el punto 7 de este enunciado.

La práctica puntúa sobre 10, el exceso de puntos que se pueda alcanzar se considerará igualmente para el cálculo de la nota media de la evaluación.

## 7.1 Criterios de evaluación

La corrección y puntuación de cada apartado se realizará atendiendo a los siguientes parámetros:

- **Funcionalidad:** Que el programa realice lo que se pide
- **Estilo de programación:** Que el código del programa sea fácilmente entendible y modificable por otras personas. Para ello deberá regirse por las directrices de la programación estructurada, la programación orientada a objetos y utilizar patrones de diseño de software.
- **Interfaz:** Que el programa que utilicen recursos gráficos que faciliten la interacción con el usuario y se usen los recursos que proporciona el entorno de desarrollo.

## 8. Instrucciones para entregar la práctica

La práctica la subiréis al servidor web en un fichero comprimido, en el que incluiréis todos los ficheros que la componen siguiendo la estructura especificada en este enunciado. Dejaréis también el enlace a vuestro git público de la práctica.

La fecha de entrega tope es el 16 de diciembre.