

# EXAMEN DWES MVC 02/03/2018

## Parte 1: Configuración básica de partida. (0 puntos)

Partimos de una instalación de Mini completamente limpia donde el proyecto se llamará examen31.dev y la base de datos 'alumnoXX' siendo XX el número que tenéis en la lista de clase. Si disponéis de archivo de configuración para la base de datos, el usuario será 'user' y la contraseña 'user'.

El proyecto utilizará la librería Plates, por lo que de partida ya puede estar instalada y si no es el caso, lo hacéis nada más empezar.

Consistirá en desarrollar una aplicación para que los usuarios registrados tengan acceso a su propia agenda de contactos. La base de datos constará al menos de la tabla Users (id, first\_name, last\_name, email, password), Contacts (id, first\_name, last\_name, phone, email, address, category\_id, user\_id), Categories (id, name, description). Donde cada contacto pertenece a un usuario y tiene una categoría (por ejemplo amigos, familia, trabajo, etc).

## Parte 2: Acceso a la sección privada. (1'5 puntos)

Creamos un login y registro de usuarios, para que cualquiera pueda registrarse en nuestra aplicación y posteriormente pueda autenticarse. Los usuarios registrados se guardarán en la tabla Users.

## Parte 3: Tablas Categories, Users y Contacts. (0 puntos)

Creamos en la Base de Datos las tablas descritas en el punto 1.

## Parte 4: Listado de la tabla Contacts. (1 puntos)

Cuando el usuario se autentique en el sistema, su página privada debe mostrar el listado de contactos, pero sólo debe ver aquellos de los que sea propietario. Aquellos que en la tabla contactos están relacionados con él. No puede tener acceso a los contactos de otros usuarios.

## Parte 5: CRUD de la Tabla Categories.

En una página privada (sólo tienen acceso los usuarios autenticados) habrá que hacer:

### Parte 5.1: Listado de la Tabla Categories. (0'75 puntos)

Mostrar en una tabla las categorías existentes, donde además de verlas, tendremos enlaces (o botones) para añadir nuevas y editarlas. No se podrán borrar las categorías.

### Parte 5.2: Creación de Categories. (0'75 puntos)

Podremos crear nuevas categorías.

### Parte 5.3: Edición de Categories. (0'75 puntos)

Podremos editar una categoría, para cambiar su nombre y/o su descripción.

## Parte 6: CRUD de la tabla Contacts.

En una página privada (sólo tienen acceso los usuarios autenticados) habrá que hacer:

#### **Parte 6.1: Listado de la Tabla Contacts. (1 punto)**

Mostrar un listado de los contactos, pero aparecerá en formato tabla, solo con aquellos campos más identificativos, para que el administrador pueda seleccionar desde ahí aquellas acciones que quiera hacer con ellos (editar o borrar). También debe haber un botón para crear contactos.

#### **Parte 6.2: Creación de Contactos. (1 punto)**

Habrà que crear un formulario para recabar los datos necesarios. El campo ID no se pide pues lo asigna la BD. La categoría será un campo SELECT con los datos de la tabla Categories.

#### **Parte 6.3: Edición de Contactos. (1 punto)**

Podremos editar el contacto seleccionado.

#### **Parte 6.4: Borrado de Contactos. (0'5 puntos)**

Podremos borrar el contacto seleccionado.

#### **Parte 7: Buscador. (1'75 puntos)**

Dispondremos de un buscador, tanto en la página pública como privada de los listados, que nos permita escribir el nombre de un contacto. Mostraremos los contactos (siguiendo el mismo formato de presentación de la página en la que nos encontramos) que contengan el nombre a buscar.

#### **OBSERVACIONES**

Desde cualquier página que creemos, debemos mostrar un menú que nos permita acceder a las diferentes partes de la página que sean accesibles según esté logueado o no. El menú debe mostrar sólo lo que se pueda hacer dependiendo de si está logueado o no.

Se valorará tanto la funcionalidad de lo producido como la calidad del software generado. Se utilizará todo aquello que resulte necesario relacionado con los temas vistos en clase.

El código debe estar adecuadamente comentado. Si no diera tiempo, me reservo la posibilidad de citaros individualmente para una entrevista personal donde explicaréis el código desarrollado en el examen. Es conveniente que mantengáis una copia del examen presentado.

Se subirá a Moodle un archivo comprimido de una carpeta llamada alumnoXX siendo XX el número correspondiente a cada uno y la base de datos se llamará alumnoXX siendo XX el mismo número. Todas las referencias a archivos deben ser dinámicas.

También tendréis que incluir en el archivo a subir una exportación completa de la BD que se puede hacer con phpmyadmin o adminer y será un archivo con extensión sql. No olvidéis indicar que tiene que crear la BD además de las tablas y los datos.