

# Práctica: Construcción del portal y trabajo teórico

(versión para trabajo individual)

Fecha de entrega: 12 de junio de 2016

## Parte I – Implementación del portal

La práctica consiste en crear un prototipo funcional de la web de valoraciones de restaurantes propuesta en la práctica anterior, contemplando la siguiente funcionalidad:

### *Portada inicial*

La portada contiene una cabecera con el logotipo de la web de valoraciones y las opciones para navegar entre las categorías (las categorías puedes elegir las tú mismo y dejarlas fijadas en la base de datos). La cabecera también tiene que contener los controles necesarios para iniciar sesión o desconectarse si la sesión está iniciada.

Debajo de la cabecera, en un formato a dos columnas, se incluyen los restaurantes destacados. Para cada uno, se incluye una foto pequeña, el nombre y la categoría.

### *Vista detallada de un restaurante*

En la vista detallada se muestra el nombre, seguido de una foto, el tipo de cocina, precio medio, dirección y una reseña del restaurante. Después se muestran los comentarios de los usuarios.

**Importante:** *No es necesario desarrollar la parte de los comentarios de los usuarios.*

### *Vista de los restaurantes de una categoría*

Al seleccionar una categoría se muestra una lista con todos los restaurantes de esa categoría. En esta vista, se muestran los restaurantes de forma compacta, usando una tabla de dos columnas.

- La primera columna contiene la foto.
- La segunda columna contiene un único párrafo con el nombre en negrita y la dirección.
- Tanto la foto como el título son un enlace a la descripción detallada.

### ***Funciones del administrador***

Tras hacer login, el administrador tiene que tener las pantallas necesarias para hacer dos funciones:

- Gestión de críticos/editores de restaurantes: Permite gestionar la lista de críticos/editores de la web. Las funciones posibles son añadir, modificar y eliminar editores / críticos.
- Gestión de la portada: Permite seleccionar qué restaurantes y en qué orden deben aparecer en la portada.

Las funciones del administrador se proponen como ampliación opcional.

En caso de no desarrollarlas, se pueden añadir los posibles editores directamente a la base de datos. En tal caso, debes indicar en un archivo readme.txt cuáles son los usuarios y contraseñas de los editores.

### ***Funciones del crítico / editor***

Tras hacer login, el crítico /editor tiene que tener las pantallas necesarias para gestionar los restaurantes:

- Ver todos los restaurantes.
- Editar un restaurante existente.
- Añadir un nuevo restaurante, con todos los datos necesarios.

## **Requisitos para la implementación**

En principio el formato es libre, aunque se deben cumplir una serie de requisitos obligatorios.

### ***Estilo del código***

Se valorará especialmente la limpieza del código HTML y PHP, así como la correcta organización de los archivos que componen el sitio web.

### ***Arquitectura de la aplicación***

Se valorará la arquitectura razonada y comprensible del sistema. En especial, se valorará que se distinga entre scripts PHP dedicados a generar vistas y scripts de apoyo (definiciones de clases, bibliotecas de funciones de apoyo, abstracción de la base de datos, etc.).

### ***Código PHP o JavaScript***

Se puede añadir cualquier elemento de Javascript, así como usar bibliotecas o frameworks de código libre siempre que su uso esté justificado y bien documentado.

### ***Estructura de la base de datos***

Todos los datos del sistema se deben almacenar en la base de datos. Se contemplarán, como mínimo, las siguientes tablas y campos (puedes añadir más):

- Editores: Contiene el nombre real (para humanos), el nombre de usuario (para hacer login) y la contraseña almacenada de forma segura.
- Categorías: Contiene el nombre real de la categoría y una clave numérica.
- Restaurantes: Cada registro de esta tabla contiene los datos de un restaurante. Incluye el nombre, año de apertura, dirección, la ruta de la imagen y el crítico/editor de la entrada.

### ***Parte opcional: publicación de la web***

Habitualmente las webs no se ejecutan en local, sino que se almacenan en un servidor. Aunque no sea obligatorio, se valorará como extra (hasta 1 punto) la publicación de la página en algún servidor accesible desde Internet.

En este caso, junto con la web debes indicar la URL en la que se puede consultar la web publicada.

## Parte II – Trabajo teórico

Para compensar el 10% de la nota de trabajo en clase se puede entregar un trabajo teórico de entre 20 y 30 páginas analizando a fondo alguna tecnología web, explicando posibles casos de uso, ejemplos de aplicación y un breve tutorial sobre su uso.

La siguiente tabla describe algunas posibles tecnologías sobre las que basar el trabajo, pero es posible escoger otra distinta.

Bootstrap	<a href="http://getbootstrap.com/">http://getbootstrap.com/</a>
jQuery	<a href="http://jquery.com/">http://jquery.com/</a>
Node.js	<a href="http://nodejs.org/">http://nodejs.org/</a>
Symfony	<a href="http://symfony.com/">http://symfony.com/</a>
Zend	<a href="http://framework.zend.com/">http://framework.zend.com/</a>
Foundation	<a href="http://foundation.zurb.com/">http://foundation.zurb.com/</a>

## Entrega

La práctica deberá entregarse antes del final del viernes 12 de junio, a través del Campus Virtual. Debe entregarse un único archivo comprimido con todos los elementos de la web y una memoria explicativa (en PDF) que explique el funcionamiento de la aplicación.

El trabajo teórico se deberá incluir también dentro del archivo comprimido.

**También debe entregarse un script para crear la base de datos (vacía) y otro para poblarla con datos de prueba.**

El nombre del archivo comprimido debe ser <Nombre> -Final.zip