Práctica Evaluable II "Adaptación de la web de la red social"

Documentación de la práctica

Juan José Jiménez García 3º especialidad Sistemas de Información

Introducción

Este documento contiene la **documentación del sitio web realizado para la segundapráctica evaluable** de la asignatura Programación Web, del curso 2016-2017. En él, se detallarán los principales aspectos de la elaboración de dicha práctica, mostrando la arquitectura de la web para su funcionamiento, así como detalles sobre las implementaciones de diversos aspectos de la misma.

Consideraciones previas

Navegador recomendado

Esta práctica ha sido realizada empleando **HTML5** y **CSS3**, por lo que se requiere un navegador compatible con dichos lenguajes. En concreto, se ha desarrollado la práctica con la última versión disponible del navegador **Google Chrome**, por lo que se recomienda visualizar la página web con dicho navegador, especialmente con la versión 57 del mismo, como mínimo.

Navegadores soportados

Se ha testeado la práctica, además de con Google Chrome, con **Mozilla Firefox** (versión 52 mínimo). Aunque estructuralmente la página web no presenta errores ni desajustes importantes a la hora de mostrar los contenidos, se han visto algunos problemas visuales que no afectan a la experiencia de uso de la página web, pero pueden ser consecuencia de la utilización de ciertas tecnologías que se detallarán más adelante.

También se ha probado la página web en la versión móvil de Google Chrome. Se recomienda tener la última versión disponible en las tiendas de aplicaciones para una correcta visualización de la práctica. La página web **NO** se ha testeado con los navegadores Microsoft Edge, Internet Explorer o Safari, así como las versiones móviles de los mismos.

Arquitectura de la web

Diagrama del modelo de datos

En la siguiente imagen se puede ver un diagrama que detalla la estructura de la base de datos de la web. Partiendo de este diagrama se han elaborado los modelos, algo de lo que hablaremos más adelante.





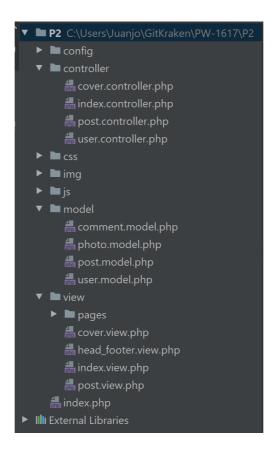




La estructura de tablas es muy sencilla, suficiente para albergar la funcionalidad básica de esta red social. Existen relaciones entre las diferentes tablas, como por ejemplo, los comentarios, que poseen el ID tanto de la publicación a la que pertenecen como el del usuario que escribió dicho comentario.

Arquitectura de clases de la web

Dicho esto, la web consta de la siguiente estructura, que podemos ver en la siguiente imagen.



Para esta práctica, he querido seguir el patrón de diseño **Modelo-Vista-Controlador**. El funcionamiento del patrón es el siguiente:

El archivo index.php representa el controlador maestro, o **controlador frontal**, el cual se encarga de recibir todas las peticiones y diferentes solicitudes que pueda hacer el usuario conforme navega por la web. Dependiendo de la URL indicada, llamará al correspondiente controlador de todos los que hay disponibles en la carpeta **controller**.

Los controladores atienden a las peticiones recibidas y en base a ello, ejecutan una u otra función de las disponibles. El **modelo** es el que se relaciona siempre con los datos. En este caso, es el que se conecta a la base de datos para conseguir la información que se le pide. El controlador NUNCA se pregunta de dónde vienen los datos, el solo los pide y los recibe. Todos los modelos están disponibles en la carpeta **model**.

La **vista** se encarga de recibir los datos (tampoco le interesa ni sabe de dónde vienen) y con ellos crea la página web. He utilizado los mismos archivos que en la primera práctica, solo que los he separado en diferentes archivos (disponibles en la subcarpeta **pages** de la carpeta **view**). De esta forma ahorramos escribir código repetido porque lo podemos reutilizar.

El patrón sigue siempre la misma mecánica:

"El controlador recibe la orden de generar una página. Para ello, identifica qué es lo que se quiere, y pregunta al modelo correspondiente para que le proporcione los datos necesarios. Una vez recogidos los datos que el modelo aporta, llama a la vista para que cree la página web usando los datos que el controlador le proporciona".

Como vemos, el controlador en el patrón **MVC** hace de mediador entre el **modelo** y la **vista**. Considero que el uso de este patrón es innovador y diferente a como se solían programar hasta ahora las páginas web, ya que facilita mucho la programación y sobre todo, hace más legible el código. Entre otras mejoras, permite una mayor escalabilidad, y se adapta mejor a las circunstancias del servidor donde nos alojamos, ya que por ejemplo, podríamos necesitar una forma diferente de obtener los datos, y para ello **solo tenemos que cambiar el modelo**, y no los controladores ni las vistas, lo que nos facilita los cambios.

Validación de formularios

Los formularios están validados en el lado del cliente mediante JavaScript. Para ello, los formularios llaman a las funciones disponibles en el archivo **functions.js** de la carpeta **js**, las cuales comprueban los datos insertados y en caso de error, informan al usuario de que existe un problema con los datos insertados y no deja continuar el proceso. Los formularios comprueban que los campos no están vacíos (aunque eso también se comprueba con HTML5 con la propiedad **required**), y que determinados campos tienen longitudes mínimas y/o máximas).

Aspectos innovadores de la práctica

Formato de fechas de las publicaciones y comentarios

Como podemos ver en la imagen, las fechas de los comentarios y las publicaciones siguen un formato que indica hace cuánto que se introdujo el elemento en cuestión. La función que logra esto está disponible en **config/functions.php** y lo que hace es recibir la hora y fecha en el formato de MySQL y adaptarla al formato que se puede apreciar en la imagen.



Patrón MVC

Ya se ha explicado anteriormente, pero quería dejarlo aquí indicado porque considero que usar patrones de diseño a la hora de crear software ayuda bastante a resolver problemas y facilita mucho la programación.