

Prueba LAMP Senior Developer

Miguel Ángel Ortiz Iglesias

29 de octubre de 2013

Índice

1. Introducción	1
2. Creación de un blog	1
3. Análisis y diseño formal	2
3.1. Análisis formal	2
3.2. Diseño formal	2
3.2.1. Diseño de la base de datos	2
3.2.2. Diseño de las webs	2
4. Solución Propuesta	2
4.1. Creación de la base de datos	2
4.2. Creación del modelo de la base de datos	2
4.3. Ejemplos de consultas	3
4.4. Ejemplos de inserciones	3
4.4.1. Capturas de pantalla	3
5. Alcance y Roadmap	3
5.1. Roadmap	3
6. Puesta en producción	4

1. Introducción

El presente documento pretende ser uno de los requisitos que imponen en la prueba de capacidades para desarrollador LAMP.

El documento está dividido en una pequeña definición del problema, ver capítulo 2, un análisis del problema, ver capítulo 3, la solución que se propone, ver capítulo 4 y una serie de posibles mejoras, ver capítulo 5.

2. Creación de un blog

La prueba de capacidad consiste en la creación de un blog con detección de móvil (Marca/Modelo) que accede. El sistema tendrá una parte pública donde se pueden ver y crear los posts y una parte de admin donde se pueden censurar.

3. Análisis y diseño formal

En el presente capítulo se va a explicar cual ha sido el resultado del análisis así como del diseño formal al que se ha llegado.

3.1. Análisis formal

Para poder ver los post que se han publicado es necesario tenerlos guardados en algún sitio, así como saber quien lo ha publicado, y si dicho post tiene una respuesta de alguien. Es por ello que se necesita una base de datos para poder acceder a dicha información.

También será necesario disponer de diferentes webs por las que navegar.

Cada usuario registrado dispone de una parte de administración para poder controlar y censurar los diferentes comentarios que considere inapropiados y borrar en caso necesario algún post de los que ha publicado.

3.2. Diseño formal

El diseño se puede dividir en dos partes.

1. Diseño de la base de datos.
2. Diseño de las webs en las que se compone la aplicación.

3.2.1. Diseño de la base de datos

La base de datos se ha realizado en UML, tal y como se puede observar en la figura 1, y representan una primera versión de lo necesario para poder desarrollar la tarea. Cabe destacar que será necesario disponer de una segunda base de datos para poder guardar los datos de los usuarios así como de su contraseña de acceso.

3.2.2. Diseño de las webs

En las figuras 2, 3, 4, 5, 6, 7, 8, 9, se pueden observar los diferentes diseños que se han planteado para cada una de las partes de la aplicación.

4. Solución Propuesta

La solución que se propone para por la creación de la base de datos en mysql, así como de una serie de consultas inserciones y actualizaciones de la misma.

Para crear la base de datos se ha creado varios script.

4.1. Creación de la base de datos

```
1 CREATE DATABASE IF NOT EXISTS db_blog;  
2  
3 USE db_blog;
```

4.2. Creación del modelo de la base de datos

```

1 CREATE TABLE IF NOT EXISTS usuario (ID INT NOT NULL
  AUTO_INCREMENT UNIQUE PRIMARY KEY, nombre TINYTEXT, posts
  INT);
2 CREATE TABLE IF NOT EXISTS post (ID INT NOT NULL
  AUTO_INCREMENT UNIQUE PRIMARY KEY, titulo TINYTEXT,
  contenido LONGTEXT, id_usuario INT REFERENCES usuario (ID
  ) ON DELETE CASCADE);
3 CREATE TABLE IF NOT EXISTS mensaje (ID INT NOT NULL
  AUTO_INCREMENT UNIQUE PRIMARY KEY, contenido LONGTEXT,
  id_post INT REFERENCES post (ID) ON DELETE CASCADE,
  id_usuario INT REFERENCES usuario (ID) ON DELETE CASCADE)
  ;

```

4.3. Ejemplos de consultas

```

1 SELECT nombre FROM usuario ORDER BY posts;
2
3 SELECT titulo FROM post ORDER BY ID DESC LIMIT 10;
4
5 SELECT contenido FROM mensaje WHERE id_post=?;

```

4.4. Ejemplos de inserciones

```

1 INSERT INTO usuario (nombre) VALUES ('Miguel Angel');
2
3 INSERT INTO post(titulo, contenido, id_usuario) VALUES
  (?, ?, ?);
4
5 INSERT INTO mensaje(contenido, id_post, id_usuario) VALUES
  (?, ?, ?);

```

4.4.1. Capturas de pantalla

En las figuras [10](#), [11](#), [12](#), [13](#), [14](#) se pueden observar las diferentes capturas de pantalla.

5. Alcance y Roadmap

5.1. Alcance

El desarrollo que se ha conseguido hasta este momento es:

- Mostrar los últimos post publicados.
- Crear nuevo post.
- Crear respuesta a un post.

está en mostrar el contenido de los post y la posibilidad de poder crear Post así como de poder crear la respuesta a los mismos.

5.2. Roadmap

Los siguientes pasos que se pueden seguir son:

- Conseguir utilizar el framework CodeIgniter como requisito.
- Añadir una base de datos con los datos y las contraseñas de los usuarios.
- Crear la web de administración de los post y los mensajes.
- Añadir la publicidad a la aplicación.
- Crear la web de modificar los datos de los usuarios.
- Mejorar la parte de acceso a la aplicación de los usuarios.
- Crear la web de alta de usuario.
-

6. Puesta en producción

La puesta en producción sería de la siguiente manera. ¹

1. Instalar el framework CodeIgniter en el servidor de producción.
2. Configurar el framework para que apunte a los ficheros correctos.
3. Copiar los ficheros correspondientes de la aplicación.
4. Comprobar que todo es correcto.
5. Habilitar el acceso a la aplicación para poner en marcha el servicio.

¹Suponemos que el servidor ya tiene instalado el apache, mysql, php y que corre bajo un sistema Linux

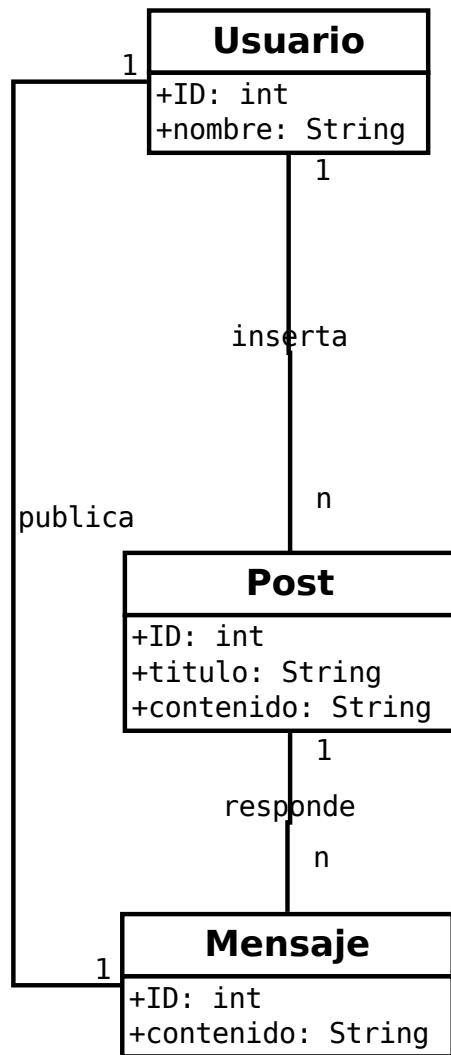


Figura 1: Modelo UML

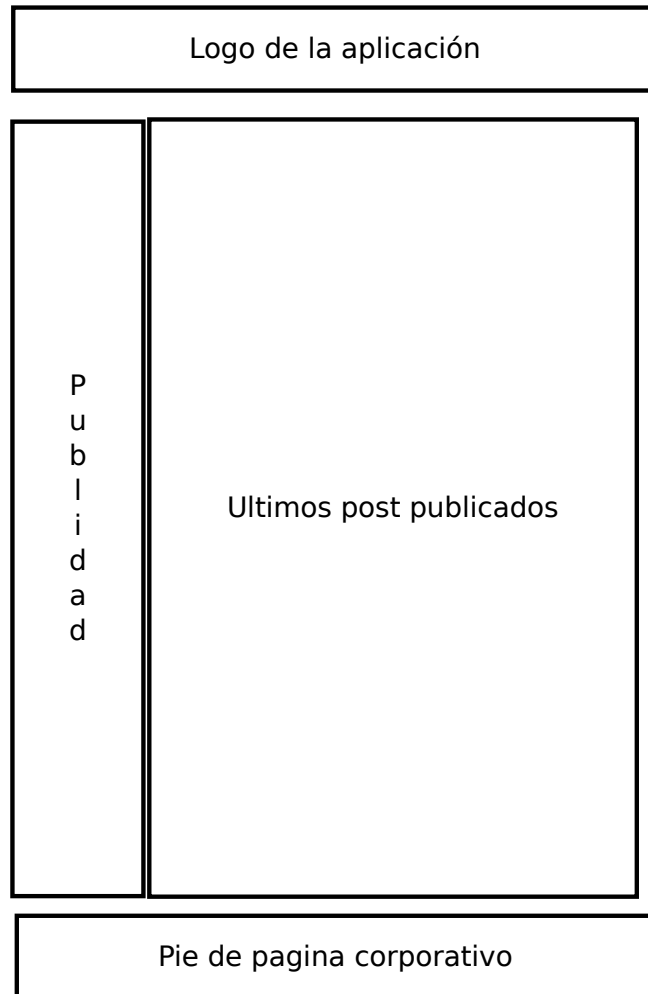


Figura 2: Diseño del home

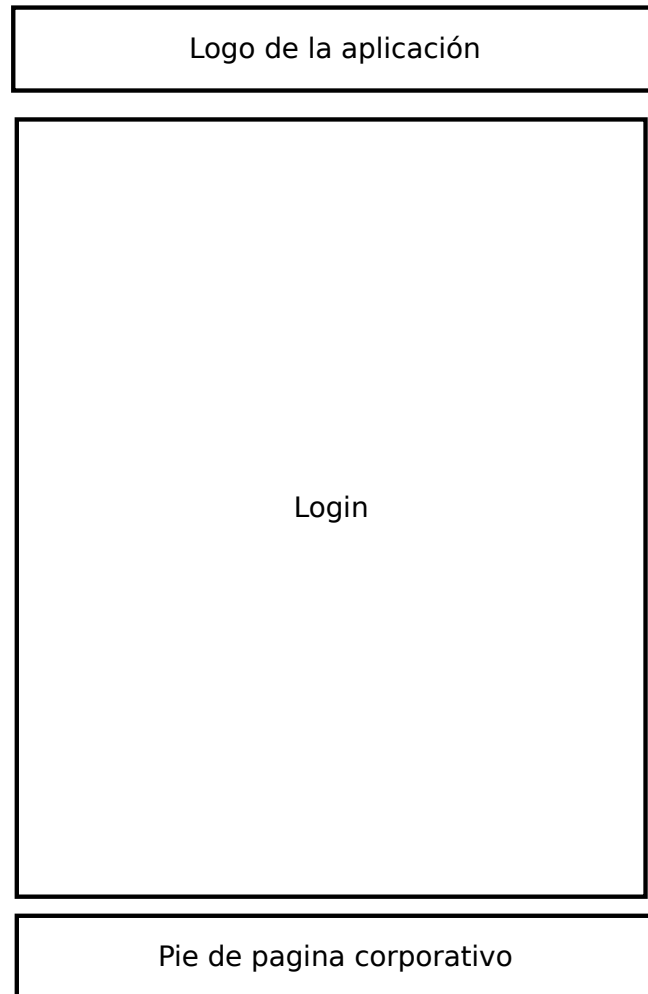


Figura 3: Diseño del login

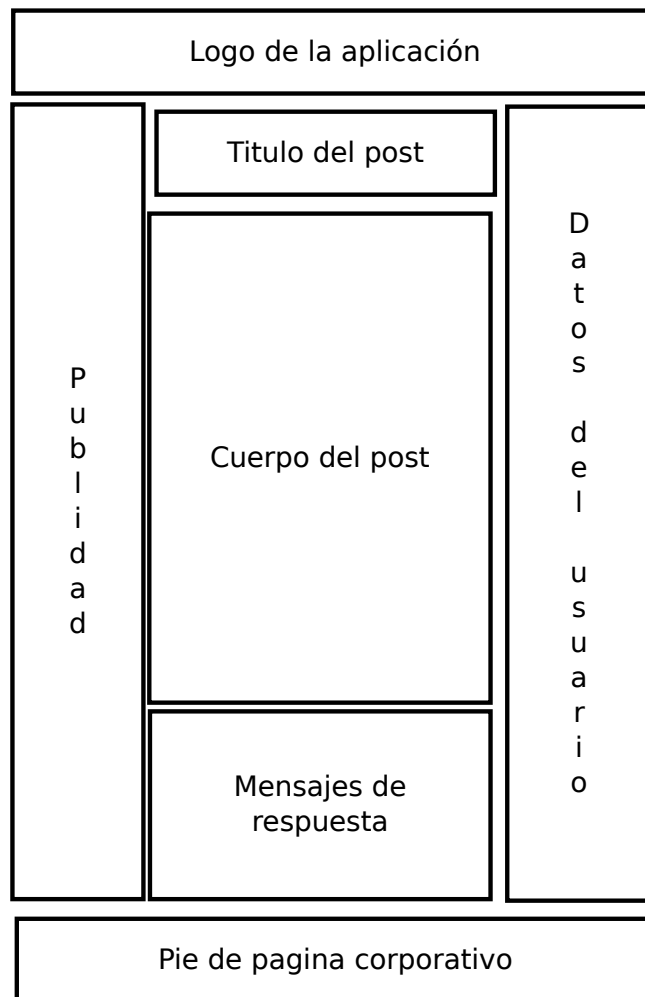


Figura 4: Diseño de mostrar un post

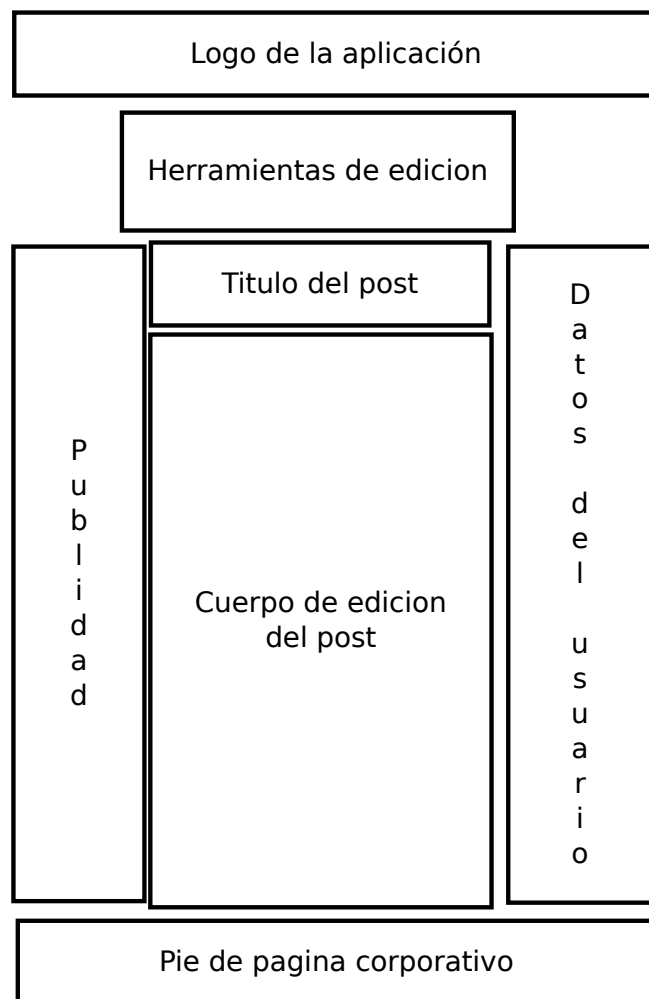


Figura 5: Diseño de la creación de posts

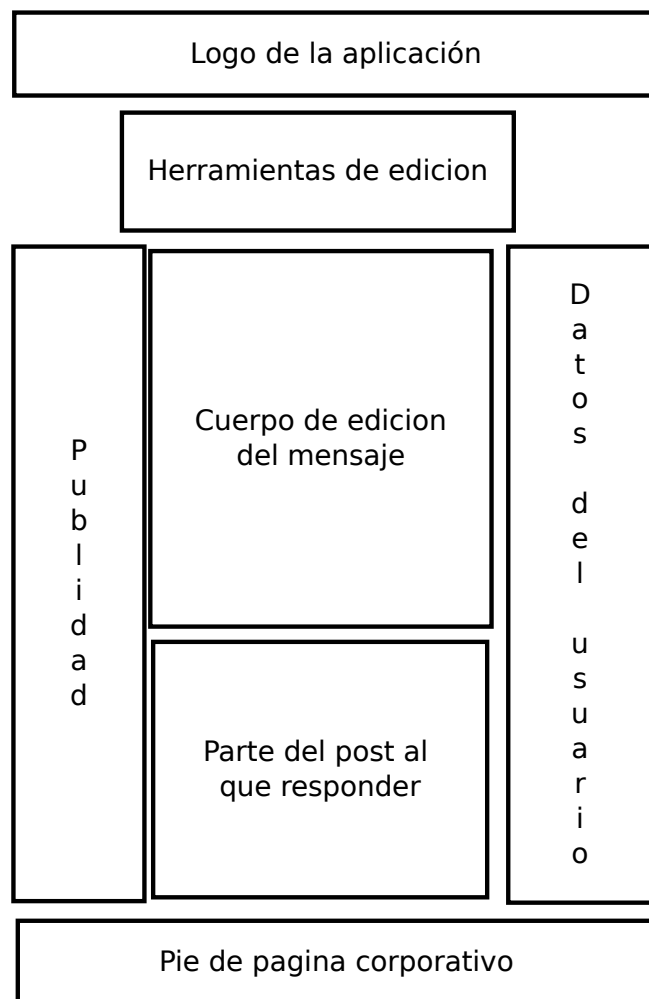


Figura 6: Diseño de la creación de mensajes

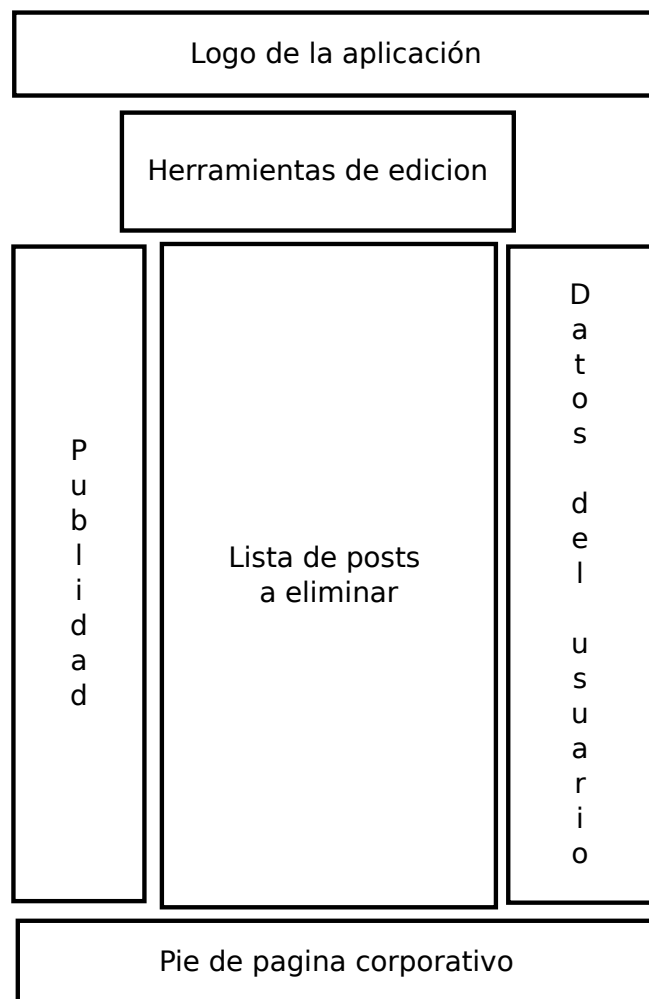


Figura 7: Diseño de la eliminación de posts

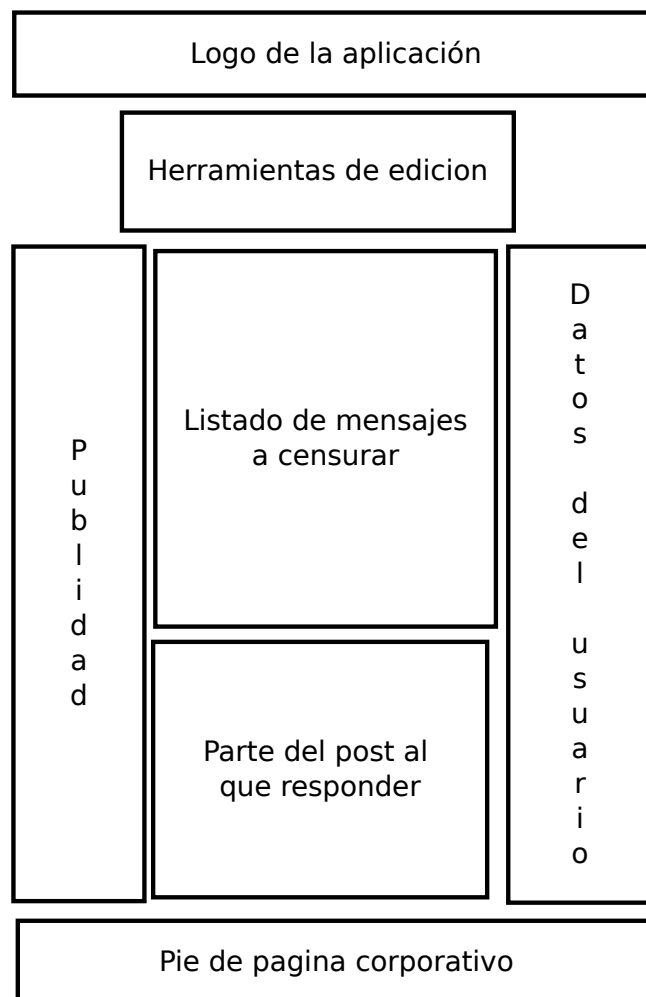


Figura 8: Diseño de la eliminación de mensajes

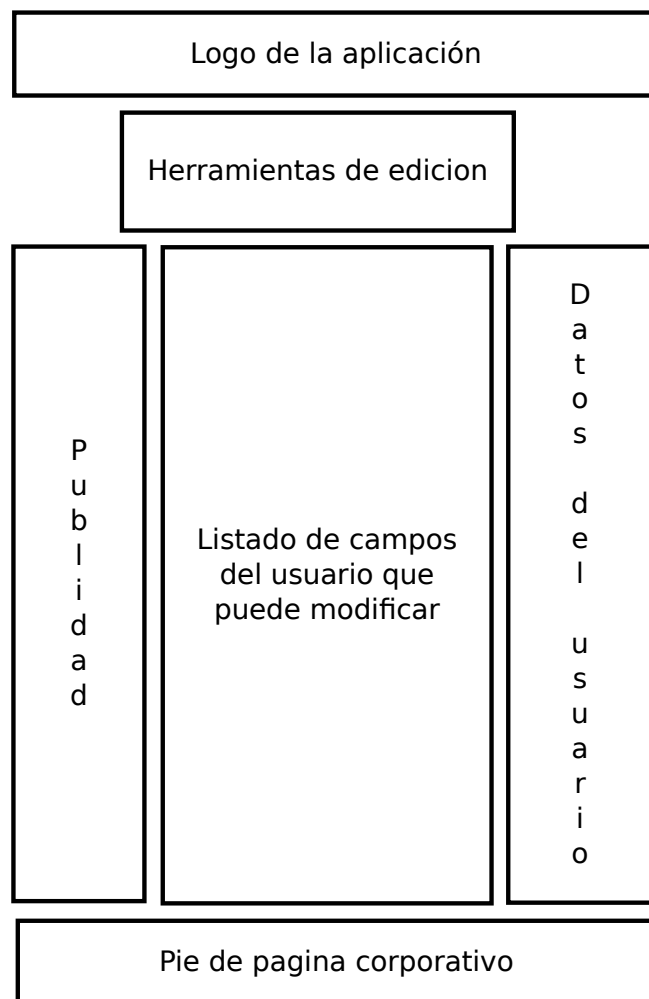


Figura 9: Diseño de la modificación de datos de usuario



Figura 10: Ejemplo de index.php



Figura 11: Ejemplo de login.php



Figura 12: Ejemplo de mostrarPost.php



Figura 13: Ejemplo de crearPost.php



Figura 14: Ejemplo de crearRespuesta.php