

# Prueba LAMP Senior Developer

Miguel Ángel Ortiz Iglesias

17 de octubre de 2013

## Índice

<b>1. Introducción</b>	<b>1</b>
<b>2. Creación de un blog</b>	<b>1</b>
<b>3. Análisis y diseño formal</b>	<b>1</b>
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
<b>4. Solución Propuesta</b>	<b>2</b>
4.1. Creación de la base de datos . . . . .	2
4.2. Creación del modelo de la base de datos . . . . .	2
<b>5. Mejoras en la aplicación</b>	<b>3</b>

## 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, 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);  
2 CREATE TABLE IF NOT EXISTS post (ID INT NOT NULL  
    AUTO_INCREMENT UNIQUE PRIMARY KEY, titulo TINYTEXT,  
    contenido LONGTEXT, id_blog INT REFERENCES blog (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)
;
```

## 5. Mejoras en la aplicación

- Añadir una base de datos con los datos y las contraseñas de los usuarios.

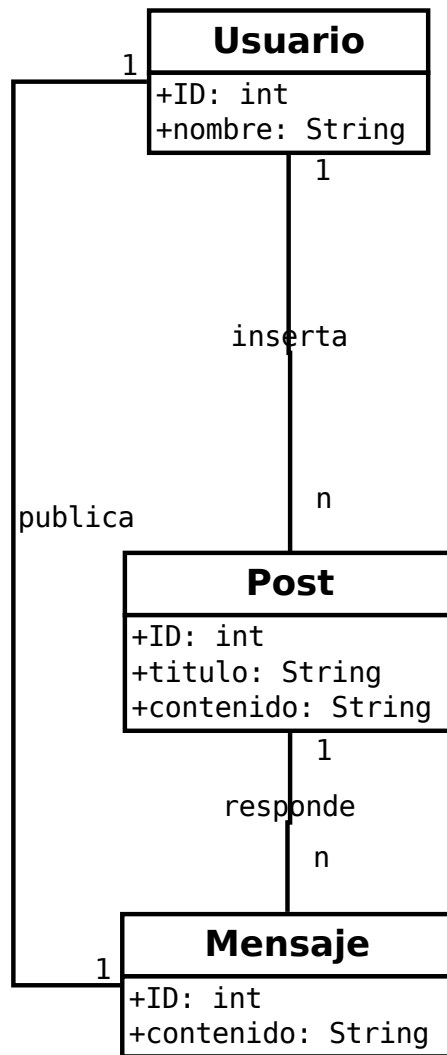


Figura 1: Modelo UML

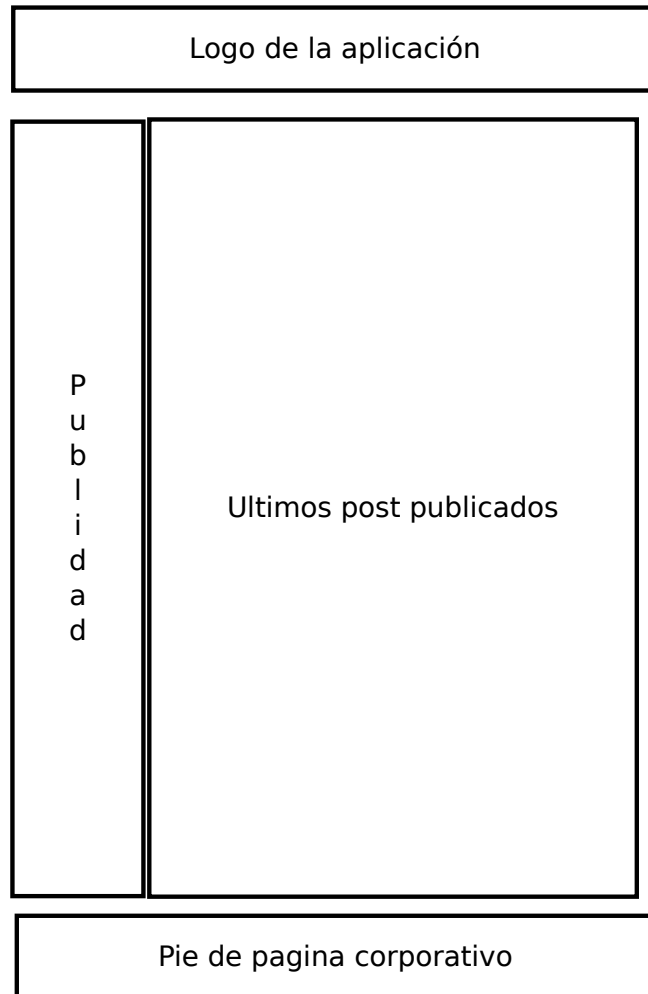


Figura 2: Diseño de la home

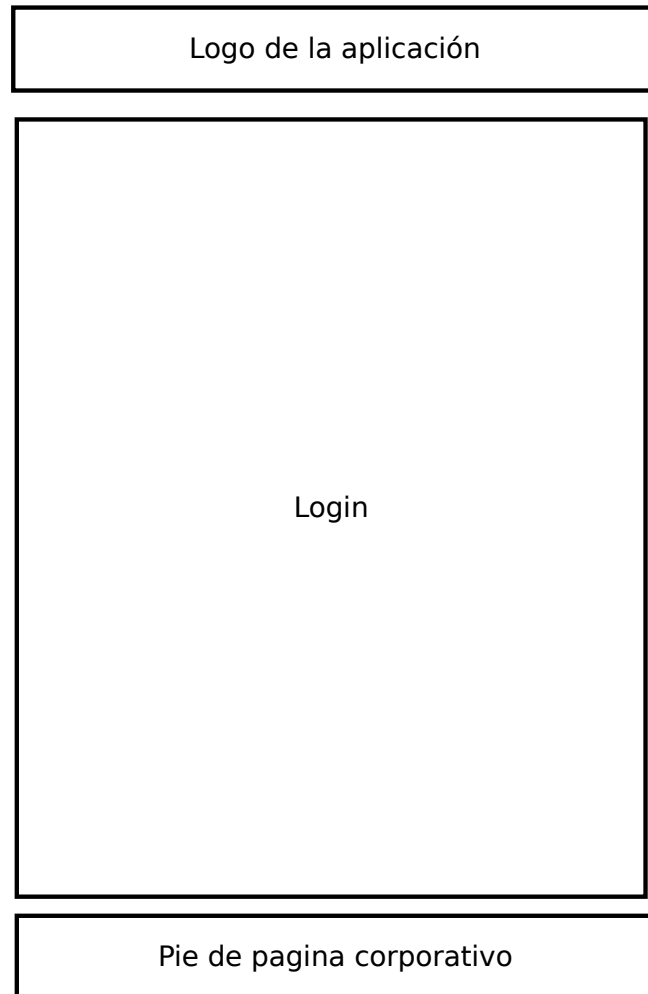


Figura 3: Diseño del login

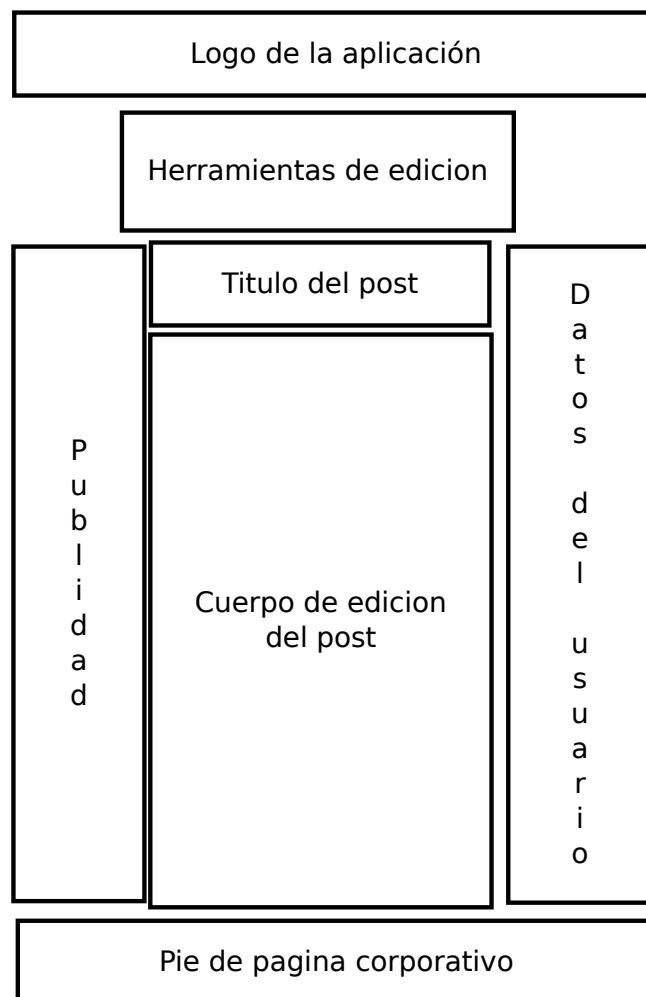


Figura 4: Diseño de la creación de posts

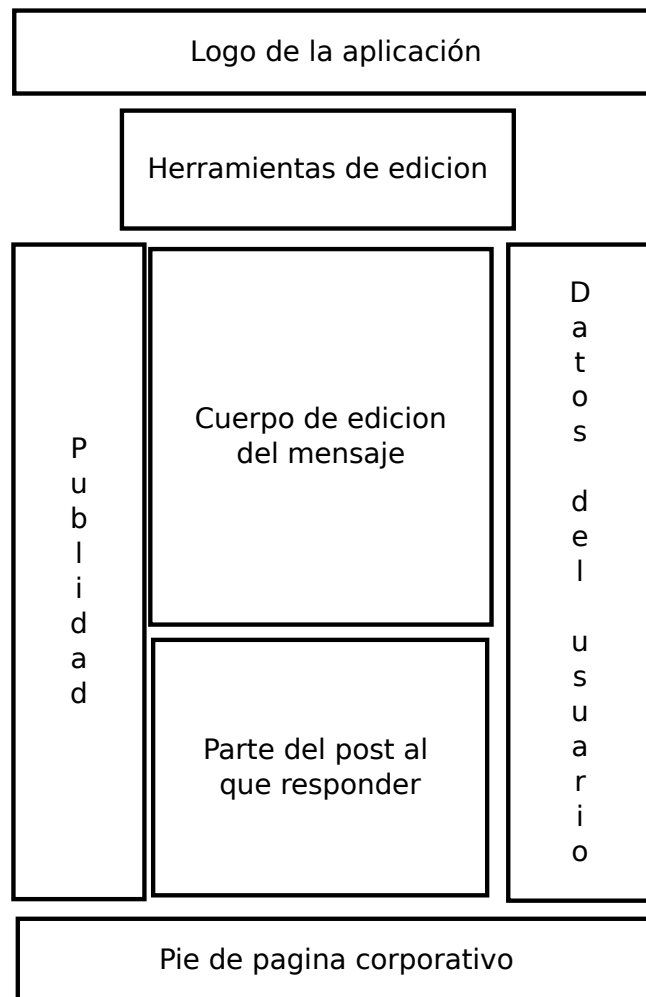


Figura 5: Diseño de la creación de mensajes



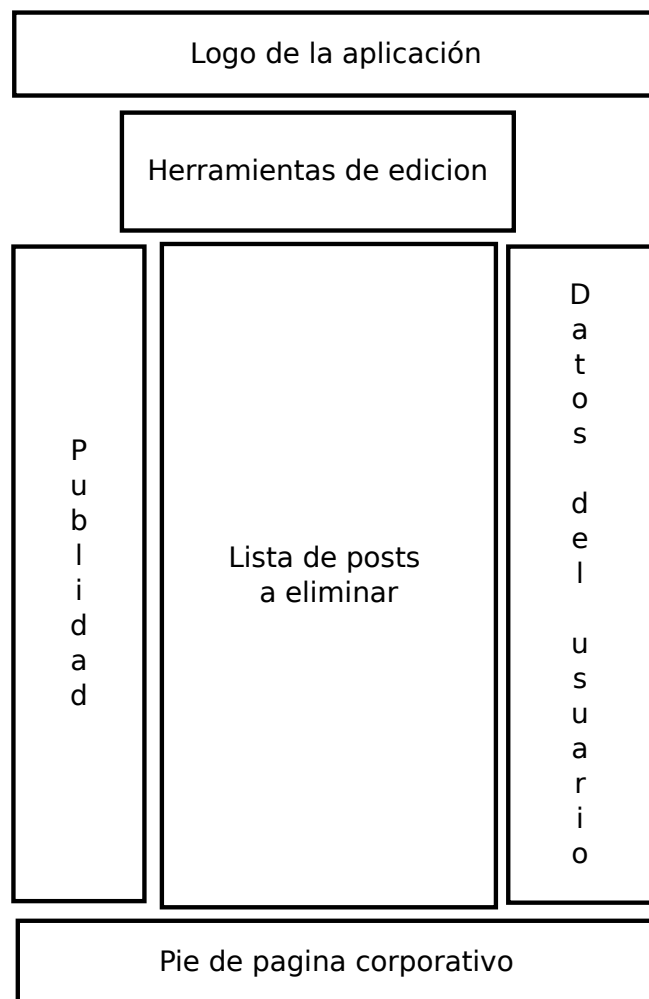


Figura 6:

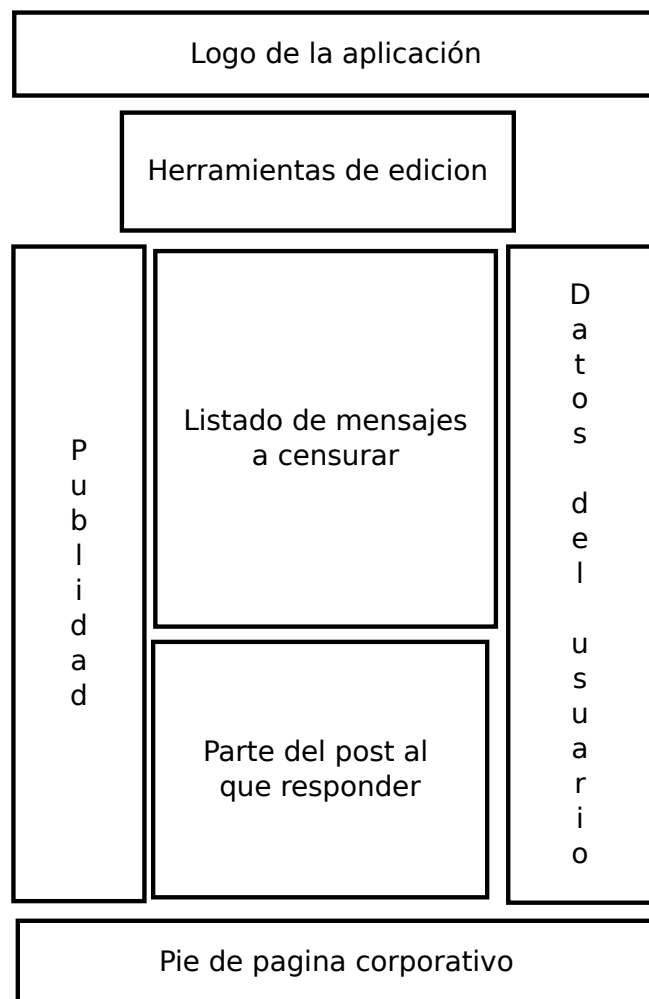


Figura 7:

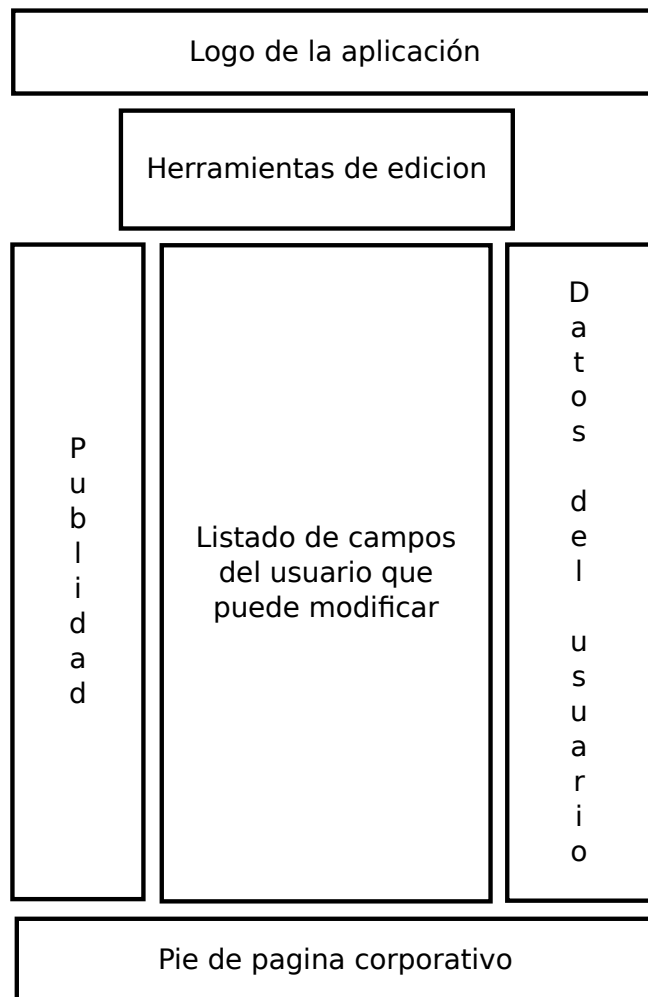


Figura 8: