

# Desarrollo de un blog

*Memoria final de CP DAW*



```
 3  namespace App\Http\Controllers\Web;
 4
 5  use Illuminate\Http\Request;
 6  use App\Http\Controllers\Controller;
 7
 8  use App\Post;
 9  use App\Category;
10
11 class PageController extends Controller
12 {
13     public function blog(){
14         $posts = Post::orderBy('id', 'DESC')->where('status', 'PUBLISHED')->paginate(3);
15
16         return view('web.posts', compact('posts'));
17     }
18
19     public function category($slug){
20
21         $category = Category::where('slug', $slug)->pluck('id')->first();
22         $posts = Post::where('category_id', $category)
23             ->orderBy('id', 'DESC')->where('status', 'PUBLISHED')->paginate(3);
24
25         return view('web.posts', compact('posts'));
26     }
27
28     public function tag($slug){
29
30         $posts = Post::whereHas('tags', function($query) use($slug){
31             $query->where('slug', $slug);
32         })
33             ->orderBy('id', 'DESC')->where('status', 'PUBLISHED')->paginate(3);
34
35         return view('web.posts', compact('posts'));
36     }
37
38     public function post($slug){
39         $post = Post::where('slug', $slug)->first();
40
41         return view('web.post', compact('post'));
42     }
43
44 }
```

Miguel Ángel Roig  
Julio 2020

Memoria final de CP DAW	1
1. Introducción	3
2. Objetivos	3
3. Contenidos	3
4. Metodología	4
5. Arquitectura de la aplicación	5
6. Plataforma de desarrollo	5
7. Planificación	6
8. Diagramas UML	7
9. Prototipos	8
10. Usabilidad	12
11. Requisitos de instalación	13
12. Instrucciones de uso	15

# Desarrollo de un blog

## 1. Introducción

Durante el curso hemos visto diferentes tecnologías que nos permiten desarrollar herramientas más complejas para la programación web que se pueden utilizar para crear diferentes tipos de proyectos, como en este caso es un blog

## 2. Objetivos

Vamos a describir los diferentes objetivos de la aplicación web:

- Adquirir los conocimientos necesarios para el desarrollo de aplicaciones web más completas
- Implantar correctamente bases de datos, backend y frontend.
- Dotar de un sistema de login para que cada usuario tenga control sobre sus publicaciones
- Disponer de un CRUD (Create, Read, Update, Delete) para los diferentes apartados del proyecto

## 3. Contenidos

La aplicación tiene un diseño sencillo para ver el contenido y poder editarlos de manera que sea intuitiva para el usuario final.

Podemos diferenciar de dos manera el formato de la web ya que por un lado tenemos la página principal y las páginas de edición que disponen de:

- Cabecera: muestra la barra de navegación para poder desplazarse por las diversas opciones
- Cuerpo: en caso de la página principal muestra el contenido y en el caso de las páginas de edición muestra un listado para poder ver, editar o eliminar el diverso contenido

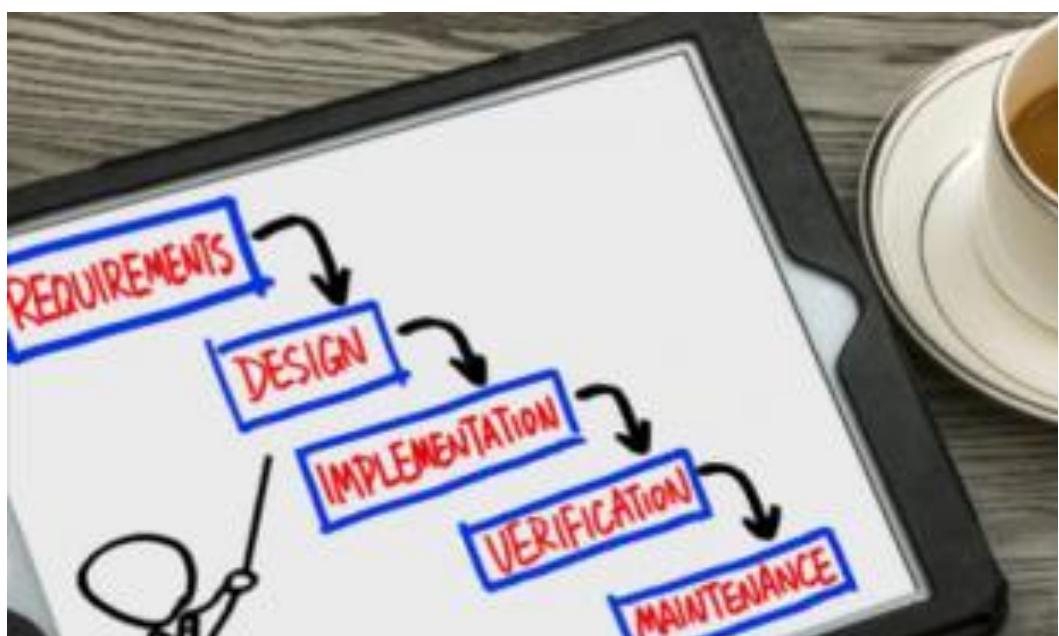
Las rutas de las que disponemos en la barra de navegación son las siguientes:

- Home: Aparece el logo de la aplicación con diferentes opciones de acceso.
- Login: En la página web principal podemos acceder con nuestro usuario para acceder a las páginas de edición.
- Register: Para poder registrarse como nuevo usuario y de esta manera poder publicar.

- Opciones de edición: una vez accedemos a la aplicación nos encontramos que en la barra de navegación nos aparecen opciones nuevas para poder editar nuestro propio contenido:
  - Etiquetas: Podremos ver, editar o eliminar las diferentes etiquetas
  - Categorías: Podremos ver, editar o eliminar las diferentes categorías
  - Entradas: Podremos ver, editar o eliminar las diferentes entradas

## 4. Metodología

Se ha utilizado una metodología en cascada a lo largo del proyecto ya que es un modelo lineal de diseño de software que emplea un proceso de diseño secuencial. El desarrollo fluye secuencialmente desde el punto inicial hasta el punto final, con varias etapas diferentes: planteamiento, iniciación, análisis, diseño, construcción, pruebas, implementación y mantenimiento.



El énfasis de la metodología en cascada se pone en la planificación de proyecto y, por tanto, antes de comenzar cualquier tipo de desarrollo es necesario que tanto la visión como el plan estén claros. Debido a que el método de cascada requiere una amplio esfuerzo de preparación previa, permite:

- Comenzar con el software con bastante rapidez.
- Estimar calendarios y presupuestos con mayor precisión.
- Lograr un nivel de satisfacción del cliente mas elevado que otros enfoques, ya desde el principio.

## *5. Arquitectura de la aplicación*

Una arquitectura de aplicaciones describe los patrones y las técnicas que se utilizan para diseñar y desarrollar aplicaciones. La arquitectura le proporciona un plan y las prácticas recomendadas que debe seguir al momento de diseñar una aplicación, de modo que obtenga una aplicación bien estructurada.

El modelo de arquitectura cliente/servidor front/end es la parte de la aplicación que interactúa con el usuario, basados en una interfaz gráfica con el usuario (GUI). El Cliente corre la aplicación que ofrece la interfaz con el usuario y back/end Es la parte no-interactiva de la aplicación. La mayor parte reside en las Bases de Datos (relacionales o no).

## *6. Plataforma de desarrollo*

Durante el desarrollo de la aplicación se han utilizado los siguientes recursos

### **Software:**

- Visual Studio Code: Es un editor de código fuente desarrollado por Microsoft para Windows , Linux y macOS . Incluye soporte para depuración , control de Git integrado, resaltado de sintaxis , finalización de código inteligente , fragmentos de código y refactorización de código.
- Laravel: es un framework que permite el uso de una sintaxis refinada y expresiva para crear código de forma sencilla, evitando el «código espagueti» y permitiendo multitud de funcionalidades. Aprovecha todo lo bueno de otros frameworks y utiliza las características de las últimas versiones de PHP. La mayor parte de su estructura está formada por dependencias, especialmente de Symfony, lo que implica que el desarrollo de Laravel dependa también del desarrollo de sus dependencias.
- PhpMyAdmin: es una herramienta que se ofrece desde los paneles de control cPanel de los alojamientos web de HOSTINET con la que podremos manejar y administrar nuestras bases de datos MySQL. Se pueden crear, eliminar, modificar bases de datos así como gestionar las tablas de las mismas.

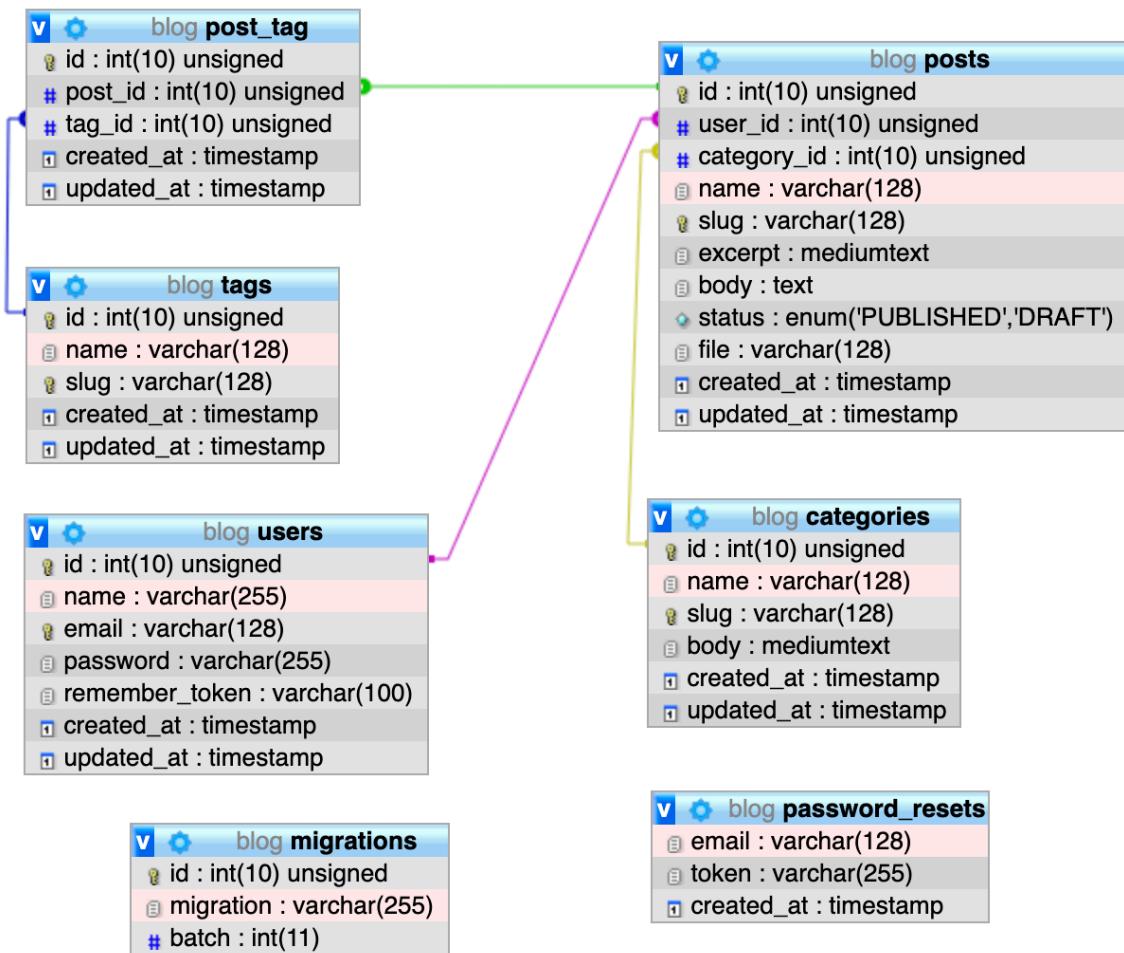
### **Hardware:**

- Portátil personal con SO MacOs.

## 7. Planificación

Tarea	Duración	Fecha de inicio	Fecha de fin
<b>Análisis/Requisitos</b>	<b>10h</b>		
Elección del proyecto	1h	15/07/2020	15/07/2020
Definición del proyecto	1h	15/07/2020	15/07/2020
Planificación del proyecto	2h	15/07/2020	15/07/2020
Elaboración del proyecto	1h	15/07/2020	15/07/2020
Elaboración memoria	2h	16/07/2020	16/07/2020
<b>Diseño</b>	<b>10h</b>		
Diagrama de clases	1h	17/07/2020	17/07/2020
Diagrama E/R	1h	17/07/2020	17/07/2020
Diagrama de casos	1h	17/07/2020	17/07/2020
Casos de uso Árbol de navegación	1h	17/07/2020	17/07/2020
Diseño de wireframes	2h	20/07/2020	20/07/2020
Diseño de base de datos	2h	20/07/2020	20/07/2020
Elaboración memoria	2h	20/07/2020	20/07/2020
<b>Implementación</b>	<b>40h</b>		
Desarrollo Backend	20h	21/07/2020	21/07/2020
Desarrollo Frontend	15h	23/07/2020	24/07/2020
Estilos	3h	27/07/2020	27/07/2020
Elaboración memoria	2h	27/07/2020	27/07/2020
<b>Entrega final</b>	<b>20h</b>		
Corrección de errores	15h	28/07/2020	30/07/2020
Finalizar memoria	5h	30/07/2020	31/07/2020

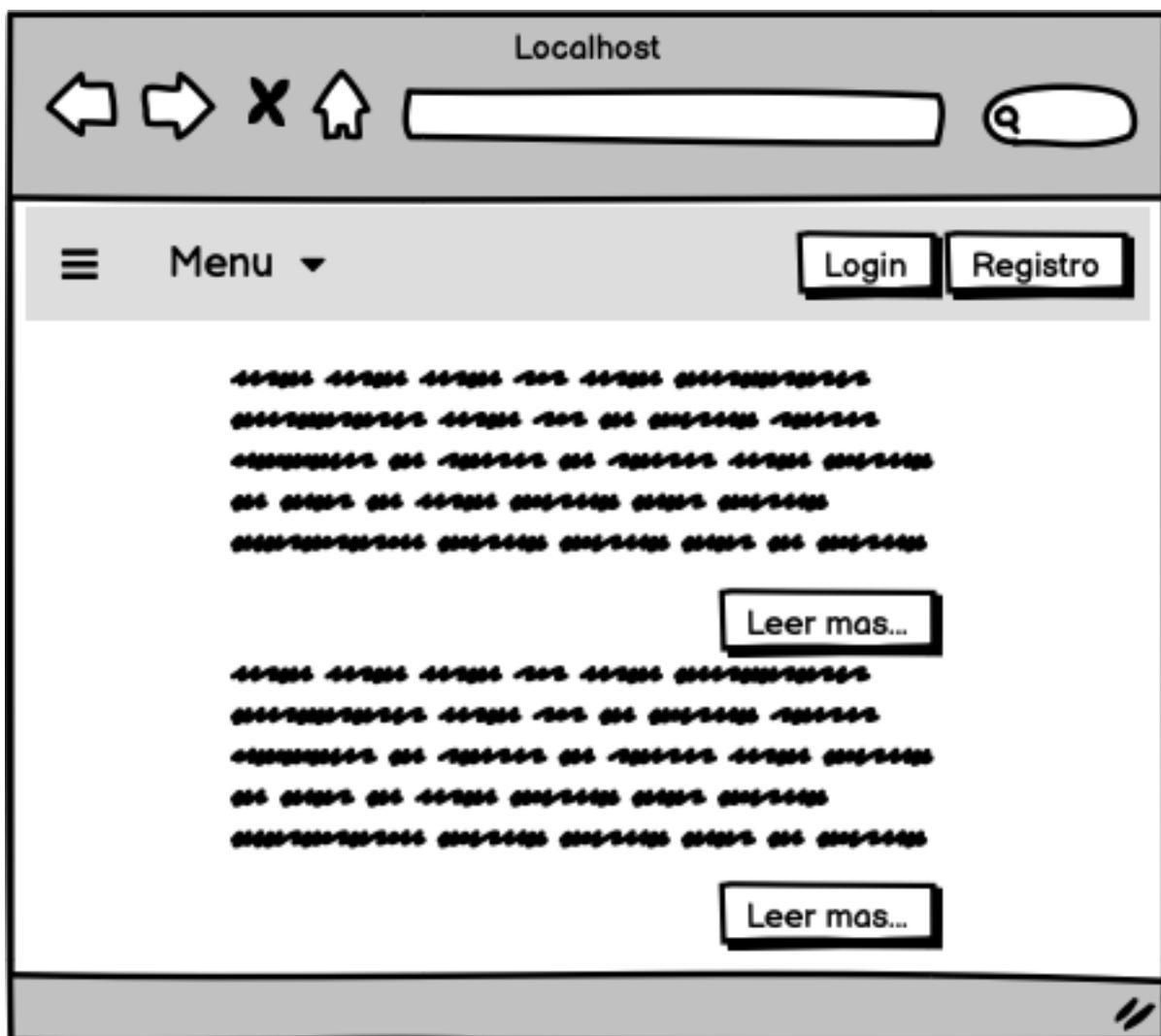
## 8. Diagramas UML



## *9. Prototipos*

Se adjuntan los wireframes con el diseño de la propuesta del proyecto de desarrollo web. Se ha utilizado la herramienta Balsamic Mockups 4 para la realización de los primeros diseños y utilizarlos como guía para el diseño final:

- Página principal:

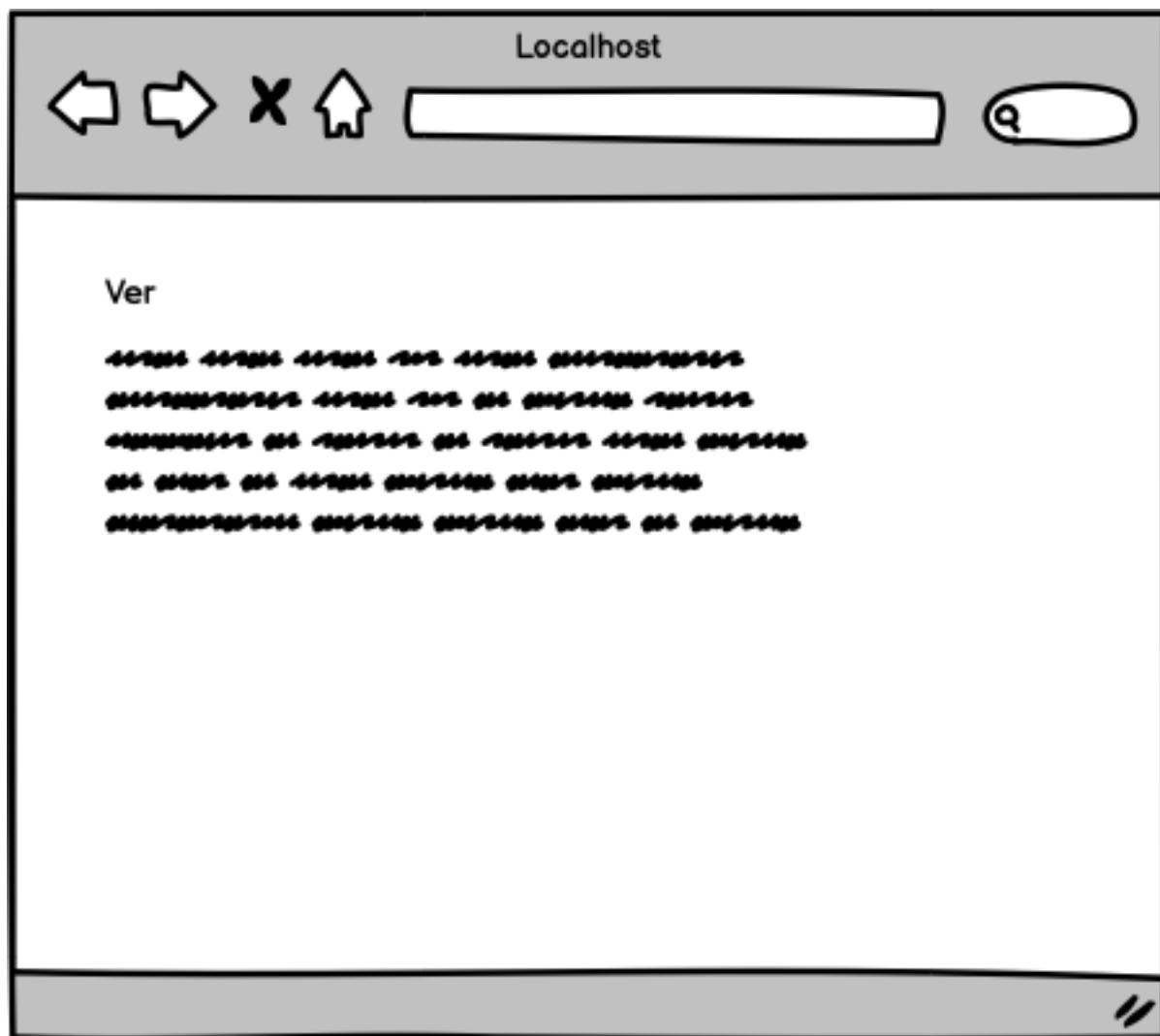


- Logeado:

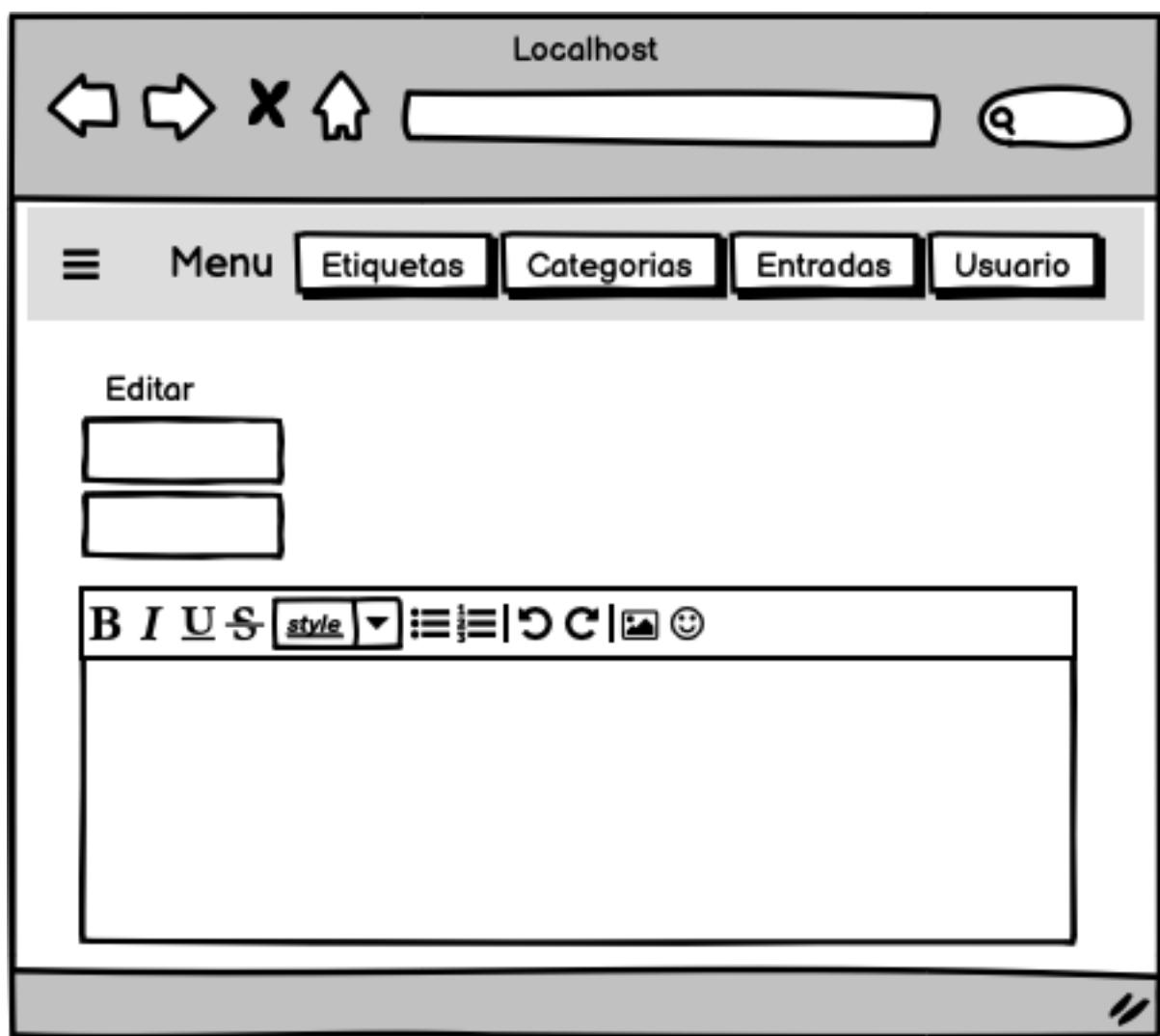
The screenshot shows a web browser window titled "localhost". The header includes standard navigation icons (back, forward, search) and a menu bar with "Menu", "Etiquetas", "Categorias", "Entradas", and "Usuario". Below the menu, the word "Lista" is displayed. A table lists six items, each with three actions: "Ver", "Editar", and "Eliminar". The first item, "Item One", is highlighted with a blue background.

Item	Ver	Editar	Eliminar
Item One	Ver	Editar	Eliminar
Item	Ver	Editar	Eliminar
Item Two	Ver	Editar	Eliminar
Item	Ver	Editar	Eliminar
Item Three	Ver	Editar	Eliminar
Item Four	Ver	Editar	Eliminar

- Ver contenido:



- Edición:

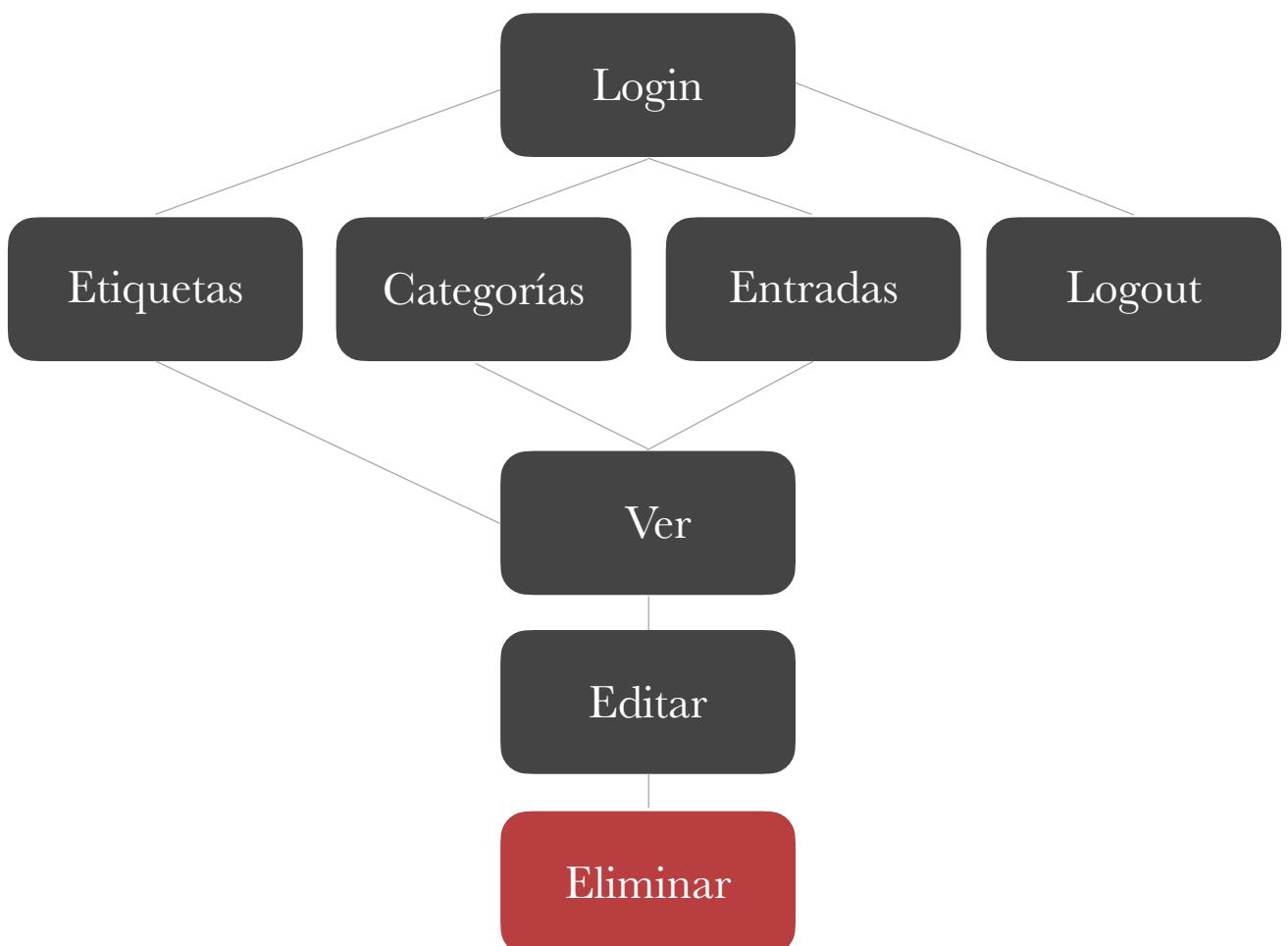


## *10. Usabilidad*

Destacamos los siguientes apartados de la usabilidad de la aplicación web de cara al usuario final:

- Una interfaz sencilla e intuitiva para que el usuario sepa moverse por la aplicación de manera intuitiva
- Informar en todo momento al usuario del estado de sus acciones al utilizar la funciones de la aplicación ver ya sea para realizar ediciones o eliminar contenido.
- El usuario final solo podrá editar sus propias publicaciones por lo que se ha tomado las medidas de seguridad necesarias para ello.

Se ha realizado el siguiente mapa de exploración de la aplicación web para resumir y simplificar el recorrido que puede realizar el usuario final:



## *11. Requisitos de instalación*

- Cliente:

- Navegador: Es necesario un navegador moderno para que no tenga conflictos a la hora de ejecutar la aplicación web y que tenga constantes actualizaciones y soporte para que no quede obsoleto como por ejemplo Google Chrome, Mozilla Firefox y Microsoft Edge:



- Servidor: Podemos optar por las herramientas Xamp, Wamp o Mamp que ofrecen incluso el gestor de base de datos. Se pueden utilizar más herramientas pero las mencionadas nos dan mayor facilidad de uso.
- Migraciones: Cuando creamos nuestras bases de datos solemos crear diagramas que nos facilitan la abstracción de como se va a almacenar nuestra información, pero la forma de llevarlo a la realidad en algún gestor de bases de datos, como por ejemplo: MySQL, SQLite, PostgreSQL, SQL Server, etc., lo más común es meternos al lenguaje de script encargado de implementar nuestra idea de la BD y ejecutar dicho script, o incluso ocupar programas más avanzados que nos sirven como interfaz para crearlas de una forma más gráfica y sin la necesidad de profundizar demasiado en el lenguaje, como Workbench o Navicat. En Laravel se lleva a otro contexto esta situación, puesto que visto de la forma tradicional si se requieren cambios en la base de datos tenemos que meternos ya sea a otro programa para cambiar el diagrama de la base o a un archivo SQL con una sintaxis usualmente complicada o difícil de leer y ejecutar los cambios para reflejarlos en el proyecto, sin embargo, con esto no contamos con un control de los cambios (control de versiones) sobre la base de datos, si necesitamos consultar un cambio anterior o de repente la solución previa o inicial era la que se necesita al momento debemos re-escribir todo otra vez, cosa que con la migraciones se

soluciona instantáneamente. Las migraciones son archivos que se encuentran en la ruta database/migrations/ de nuestro proyecto Laravel, por defecto en la instalación de Laravel 5.5 se encuentran dos migraciones ya creadas: create\_users\_table y create\_password\_resets\_table. Para crear nuestras migraciones en Laravel se usa el siguiente comando:

```
php artisan make:migration nombre_migracion
```

que nos crea el archivo limpio para escribir nuestra migración, o bien el comando:

```
php artisan make:migration nombre_migracion --create=nombre_tabla
```

que nos agrega una plantilla de trabajo básica para empezar a trabajar. Como ejemplo del curso se tomará este comando:

```
php artisan make:migration create_post_tag_table
```

El cual nos dará este resultado:

*Created Migration: 2020\_07\_21\_231303\_create\_post\_tag\_table*

Una vez creadas todas las migraciones ya podremos sincronizar nuestro proyecto con la base de datos.

## 12. Instrucciones de uso

Tenemos la página principal donde nos podemos registrar o acceder a nuestra cuenta de usuario:

The screenshot shows the homepage of a blog titled "MIKISBLOG". At the top right, there are links for "Login" and "Register". Below the header, the title "Lista de artículos" is displayed. Two article cards are visible. The first card features a woman sitting with her hands clasped, wearing a patterned dress and a necklace. The text "Accusamus laborum delectus sed." is above the image, and "Quibusdam eos vero nostrum nostrum tempora. Aut quos vel nulla quia corporis vitae placeat voluptatem. Qui ut quam ut velit minus. Porro sunt expedita dolor." is below it, with a "Leer más" link at the end. The second card shows a close-up of several dark-colored bottle caps or lids. The text "Et labore eligendi molestiae quaerat ducimus." is above the image.

Una vez registrados o logueados podremos acceder a la edición de los diferentes apartados:

This screenshot shows the same blog homepage as the previous one. The navigation bar at the top includes "MIKISBLOG", "Blog", "Etiquetas", "Categorías", "Entradas", and "Miki Roig ▾". The "Lista de artículos" section contains the same two article cards as the previous screenshot. The first card features the woman in the patterned dress, and the second card shows the close-up of bottle caps.

Cuando accedamos tanta a Etiquetas, Categorías o entradas nos aparecerá el mismo menú tanto para ver, crear, editar y eliminar etiquetas.

The screenshot shows a WordPress admin interface for managing tags. At the top, there's a navigation bar with 'MIKISBLOG' and 'Blog' on the left, and 'Etiquetas', 'Categorías', 'Entradas', and 'Miki Roig ▾' on the right. Below the navigation is a table titled 'Lista de etiquetas' (List of tags). The table has two columns: 'ID' and 'Nombre'. Each row contains a tag name with its ID and three buttons: 'Ver', 'Editar', and 'Eliminar'. The rows are numbered from 20 down to 8. The 'Eliminar' button is highlighted in red.

ID	Nombre	Ver	Editar	Eliminar
20	modi			Eliminar
19	est			Eliminar
18	voluptas			Eliminar
17	qui			Eliminar
16	harum			Eliminar
15	sint			Eliminar
14	dolore			Eliminar
13	veniam			Eliminar
12	similique			Eliminar
11	tempora			Eliminar
10	asperiores			Eliminar
9	autem			Eliminar
8	velit			Eliminar