



Daniel López Marqués |



CIFP Ponferrada Tutor: Jesús Herrero Valín  
Informática 2ºDAW 2018/2019

# ÍNDICE

<b>1</b>	<b>Estudio del problema y análisis del sistema</b>	<b>3</b>
1.1	Introducción	3
1.2	Funciones y rendimientos deseados	3
1.3	Objetivos	3
<b>2</b>	<b>Planteamiento y evaluación de diversas soluciones</b>	<b>4</b>
2.1	Estructura de la aplicación	4
2.2	Funcionamiento Robot-Motor de búsqueda	4
2.3	Búsqueda de imágenes	5
2.4	Robots.txt	5
2.5	Sistema de almacenamiento de datos	5
2.6	Alta de sitios web	5
2.7	Seguridad	5
2.8	Servidor	5
<b>3</b>	<b>Justificación de la solución elegida</b>	<b>6</b>
3.1	Diagramas UML	6
3.1.1	Diagrama de casos de uso	6
3.1.2	Diagramas de actividad	9
3.1.3	Diagrama de despliegue	10
3.1.4	Diagramas de secuencia	10
3.1.5	Diagramas de clases	12
3.2	Estructura de la aplicación	14
3.3	Funcionamiento Robot-Motor de búsqueda	14
3.4	Búsqueda de imágenes	16
3.5	Robots.txt	16
3.6	Coordinación Robot-Motor de búsqueda	17
3.7	Sistema de almacenamiento de datos	18
3.7.1	Base de datos	18
3.7.2	Ficheros	21
3.7.3	Imágenes	21
3.8	Mantenimiento	21
3.9	Alta de sitios web	22
3.10	Seguridad	22
3.10.1	Página web	22
3.10.2	Robot de indexación Java	23

3.10.3	Cifrado HTTPS.....	24
3.11	Registro de usuarios .....	24
3.12	Logs .....	25
3.13	Sistema de correo electrónico.....	25
3.14	Estructura de la web.....	26
3.14.1	Cabeceras.....	27
3.14.2	Pie de página .....	29
3.14.3	Buscador .....	29
3.14.4	Páginas principales.....	31
3.14.5	Páginas de resultados .....	33
3.14.6	Página de alta de sitios.....	35
3.14.7	Historial .....	36
3.14.8	Login y registro .....	37
3.14.9	Verificación de email y recuperación de contraseña..	38
3.14.10	Panel de administración .....	40
3.14.11	Configuración y preferencias .....	43
3.14.12	Páginas informativas .....	44
3.14.13	Páginas de errores .....	47
3.14.14	Imágenes .....	48
3.15	Hosting y dominio.....	49
<b>4</b>	<b>Modelado de la solución.....</b>	<b>50</b>
4.1	Recursos humanos.....	50
4.2	Recursos hardware .....	50
4.3	Recursos software.....	50
<b>5</b>	<b>Planificación temporal .....</b>	<b>51</b>
<b>6</b>	<b>Costes del proyecto .....</b>	<b>52</b>
<b>7</b>	<b>Conclusiones finales .....</b>	<b>53</b>
7.1	Grado de cumplimiento de los objetivos fijados.....	53
7.2	Propuesta de modificaciones o ampliaciones futuras del sistema implementado .....	54
<b>8</b>	<b>Bibliografía y webgrafía.....</b>	<b>55</b>

# *1. Estudio del problema y análisis del sistema*

## *1.1. Introducción*

Se desea desarrollar una aplicación web que permita la búsqueda en la red, a partir de la introducción de una cadena de texto en el buscador.

El desarrollo de esta aplicación web será de gran complejidad en el servidor, ya que se deberán usar algoritmos complejos para crear un robot que rastree las páginas web y las indexe en la base de datos de la aplicación; por contraposición de la parte cliente, que será lo más sencilla posible de utilizar para que sea accesible a todos los tipos de usuarios.

Para la puesta en funcionamiento y el crecimiento de la aplicación, será preciso que los creadores de páginas web se den de alta en el buscador, mediante su URL, para que su página sea rastreada e indexada por el robot, respetando el típico fichero robots.txt, y aparezca en los resultados de búsqueda.

Se implementará la creación de usuarios, permitiendo que cada usuario, entre otras cosas, pueda consultar su historial de búsqueda.

También se plantea la inclusión de otras modalidades de búsqueda, además de la búsqueda de texto.

## *1.2. Funciones y rendimientos deseados*

Desarrollar un buscador de internet completamente funcional con base de datos expandible a través de un formulario, que sea accesible para todos los tipos de usuarios.

## *1.3. Objetivos*

- Crear una aplicación web capaz de buscar entre las páginas web registradas en su base de datos.
- Permitir al usuario dar de alta nuevas direcciones en el buscador.
- Funcionamiento completamente automático para el usuario.
- Permitir el registro de usuarios para permitir el acceso a más funciones.
- Almacenar un historial de búsqueda para los usuarios registrados.
- Incluir más modalidades de búsqueda además de texto.

## 2. Planteamiento y evaluación de diversas soluciones

### 2.1. Estructura de la aplicación

Para implementar estas funciones, la aplicación estará dividida en dos partes bien diferenciadas: la página web, que contendrá el motor de búsqueda; y el programa robot que rastreará e indexará la red.

El funcionamiento de ambas partes debe estar de alguna forma “separado”, ya que el hecho de que el robot este ejecutándose no debe impedir que los usuarios visiten la página web y utilicen el motor de búsqueda con normalidad.

Para dar solución a esta estructura de aplicación, se plantean dos formas principales:

- Implementar toda la aplicación en PHP, tanto el motor de búsqueda como el robot. De este modo, ambas aplicaciones se ejecutarían con peticiones al servidor, pero el funcionamiento de una sería dependiente de la otra. Es decir, si el robot se está ejecutando, la página web debería esperar.
- Dividir la aplicación en Java y PHP, haciendo independientes el funcionamiento de cada parte. La página web y el motor de búsqueda se implementarían en PHP y el programa robot en Java SE, para ejecutarse en el servidor de forma local.

### 2.2. Funcionamiento Robot-Motor de búsqueda

Se debe tener en cuenta que un factor fundamental en esta aplicación es la velocidad de búsqueda, para ofrecer una experiencia agradable al usuario, sin grandes esperas a la hora de buscar. Para el funcionamiento del robot y el motor de búsqueda también se plantean dos soluciones:

- El robot rastrea direcciones de sitios web, en base a una lista existente de direcciones, y amplía ésta cada vez que encuentra una nueva dirección, que a su vez también rastreará en un proceso recursivo. Por su parte, el motor de búsqueda, cuando recibe un parámetro, accede a cada URL y busca ese parámetro hasta encontrar coincidencias, que mostrará por pantalla al usuario. Esto haría que el robot funcionase con mayor velocidad, pero ralentizaría el funcionamiento del motor de búsqueda.
- El robot rastrea las direcciones en busca de enlaces, pero también guarda cada palabra de la página web en una lista agrupada por cada dirección, haciendo un índice para el motor de búsqueda. El motor de búsqueda, en este caso, hace una consulta de los parámetros en dicha lista y obtiene una lista de direcciones que contienen esos parámetros, que devuelve al usuario. Esto ralentizaría el funcionamiento del robot, pero aumentaría en gran cantidad la velocidad del motor de búsqueda.

### *2.3. Búsqueda de imágenes*

También se quiere implementar la búsqueda de imágenes, para lo cual se plantea la siguiente solución:

El robot indexa, además de las palabras, las imágenes de cada dirección, y las guarda en una lista del mismo formato que la de palabras. El motor de búsqueda obtendría las direcciones correspondientes a los parámetros, y mostraría todas las imágenes de esas direcciones.

### *2.4. Robots.txt*

Es muy importante que el robot de indexación consulte el fichero robots.txt de los sitios que va a indexar. Este fichero contiene las direcciones que los robots NO deben indexar, que cada sitio web indica.

Para consultar este fichero, se plantea el uso de alguna API que facilite la tarea, ya que es un fichero complejo (indica qué User-Agent está afectado, qué zonas de la web están permitidas o prohibidas, direcciones específicas...).

### *2.5. Sistema de almacenamiento de datos*

En cuanto al sistema de almacenamiento de datos, se plantean como soluciones la utilización de base de datos en la totalidad de la aplicación, la utilización de ficheros, o la utilización de ambos, según la necesidad.

### *2.6. Alta de sitios web*

Se quiere implementar la posibilidad de que los usuarios den de alta en la base de datos una dirección de un sitio web, para ponerla a la cola para que el robot la indexe y aparezca en los resultados. Para ello se diseñará un formulario en la página web, y se añadirá, si no exista, la dirección a la lista de sitios existentes.

### *2.7. Seguridad*

Para la seguridad en la aplicación web se plantea la utilización de un framework de PHP, como Laravel, que ya proporciona las medidas básicas.

En cuanto a la parte del robot de indexación, sea en Java o PHP, hay una gran brecha de seguridad que debe solventarse. El robot de indexación podría ocasionar problemas al acceder a cualquier dirección que encuentre en la red, ya que también puede encontrarse con webs maliciosas.

Para solventar este problema, se plantea la solución de utilizar alguna API o Web Service que permita analizar una URL previamente, antes de acceder a ella.

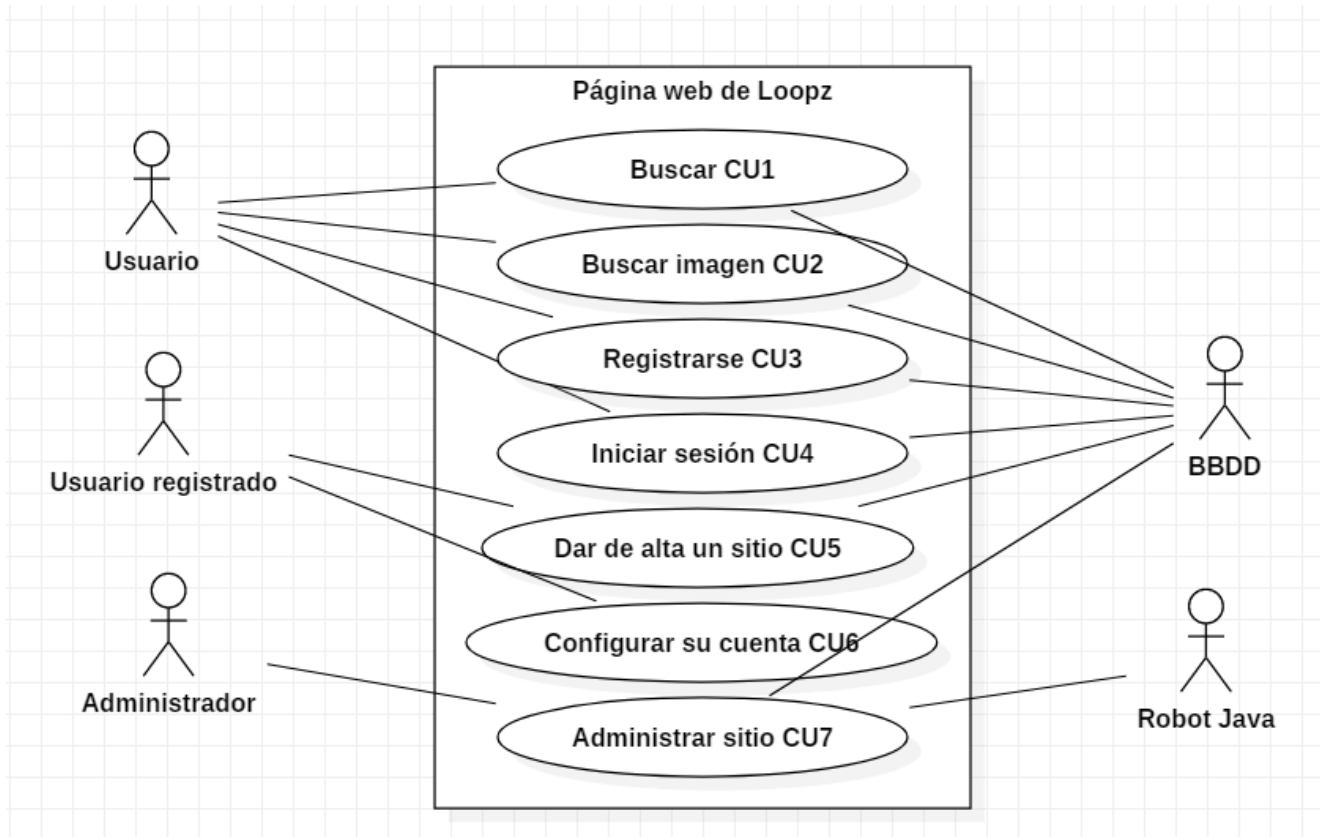
### *2.8. Servidor*

El servidor que se utilice debe tener, en principio, el permiso para ejecutar comandos desde código, ya que es necesario para hacer las peticiones a las páginas web.

### 3. Justificación de la solución elegida

#### 3.1. Diagramas UML

##### 3.1.1. Diagrama de casos de uso



<b>Caso de uso: Buscar texto</b>	<b>Caso de uso: Buscar imagen</b>
<b>ID: CU1</b>	<b>ID: CU2</b>
<b>Descripción:</b> Un usuario hace una búsqueda de texto en la página web	<b>Descripción:</b> Un usuario hace una búsqueda de imágenes en la página web
<b>Actores principales:</b> Usuario	<b>Actores principales:</b> Usuario
<b>Actores secundarios:</b> BBDD	<b>Actores secundarios:</b> BBDD
<b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El caso de uso empieza cuando un usuario va a hacer una búsqueda de texto</li> <li>2. Escribe lo que quiere buscar y envía</li> <li>3. El motor de búsqueda hace consultas a la base de datos</li> <li>4. La página web muestra los resultados</li> </ol>	<b>Flujo principal:</b> <ol style="list-style-type: none"> <li>5. El caso de uso empieza cuando un usuario va a hacer una búsqueda de imágenes</li> <li>6. Escribe lo que quiere buscar y envía</li> <li>7. El motor de búsqueda hace consultas a la base de datos</li> <li>8. La página web muestra los resultados</li> </ol>

<b>Caso de uso: Registrarse</b>	<b>Caso de uso: Iniciar sesión</b>
<b>ID: CU3</b>	<b>ID: CU4</b>
<b>Descripción:</b> Un usuario se registra en la página web	<b>Descripción:</b> Un usuario inicia sesión en la página web
<b>Actores principales:</b> Usuario	<b>Actores principales:</b> Usuario
<b>Actores secundarios:</b> BBDD	<b>Actores secundarios:</b> BBDD
<b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El caso de uso empieza cuando un usuario va a registrarse.</li> <li>2. Introduce su email.</li> <li>3. Introduce su nombre.</li> <li>4. Introduce contraseña personal.</li> <li>5. El sistema envía los datos a la base de datos.</li> </ol>	<b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El caso de uso empieza cuando un usuario va a iniciar sesión</li> <li>2. El usuario introduce sus datos</li> <li>3. Si son correctos <ol style="list-style-type: none"> <li>a. El usuario inicia sesión</li> </ol> </li> <li>4. Si son incorrectos <ol style="list-style-type: none"> <li>4.1 Volver al paso 2</li> </ol> </li> </ol>

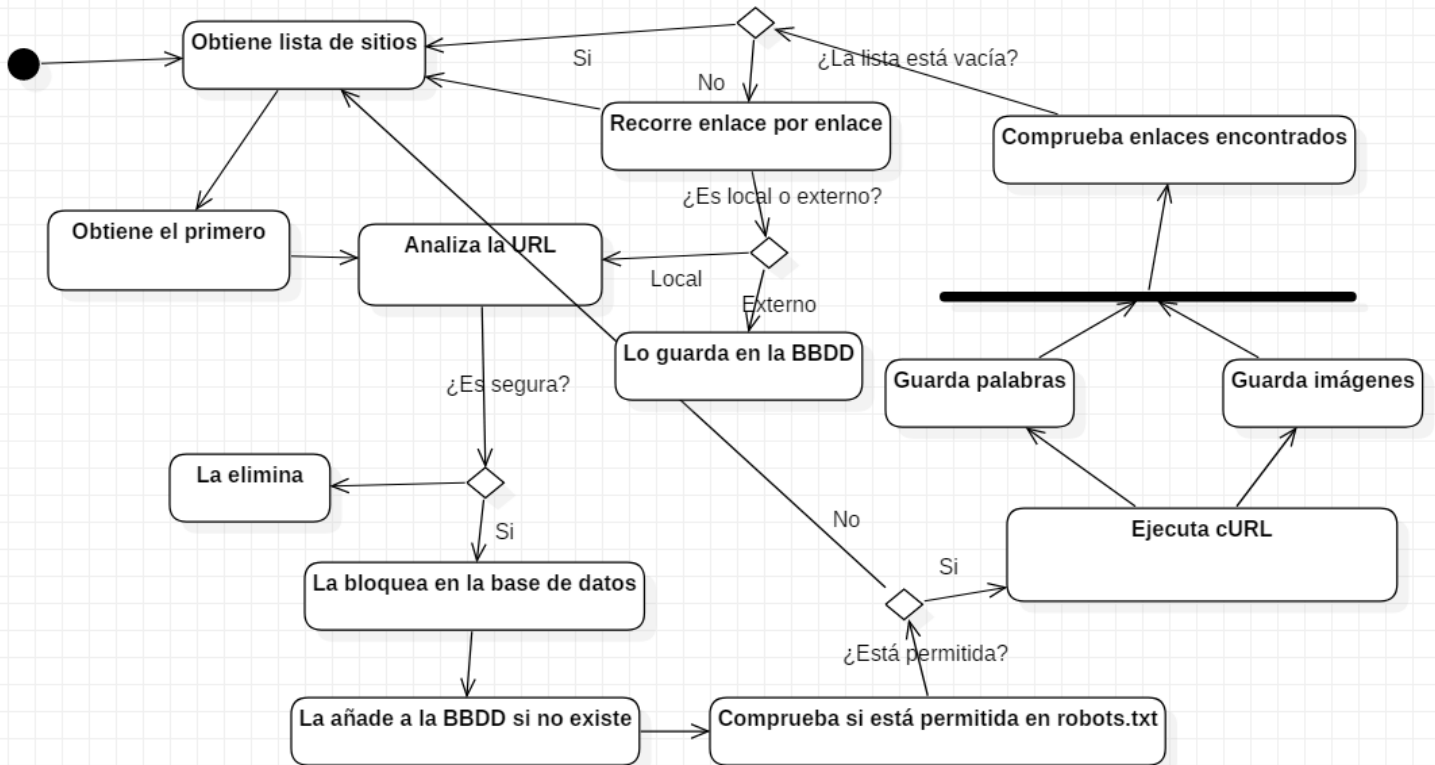


<b>Caso de uso: Dar de alta un sitio</b>	<b>Caso de uso: Configurar su cuenta</b>
<b>ID: CU5</b>	<b>ID: CU6</b>
<b>Descripción:</b> Un usuario registrado da de alta un sitio en la base de datos	<b>Descripción:</b> Un usuario registrado configura su cuenta
<b>Actores principales:</b> Usuario	<b>Actores principales:</b> Usuario
<b>Actores secundarios:</b> BBDD	<b>Actores secundarios:</b> BBDD
<b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El caso de uso empieza cuando un usuario registrado va a dar de alta un sitio web</li> <li>2. Introduce la dirección y la envía</li> <li>3. El sistema, tras hacer la validación, registra en la base de datos</li> </ol>	<b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El caso de uso empieza cuando un usuario de uso entra en su panel de configuración.</li> <li>2. En él puede: <ol style="list-style-type: none"> <li>2.1 Cambiar nombre</li> <li>2.2 Cambiar contraseña</li> <li>2.3 Cambiar email</li> <li>2.4 Cambiar preferencias</li> <li>2.5 Eliminar cuenta</li> </ol> </li> <li>3. El sistema modifica los datos en la base de datos.</li> </ol>

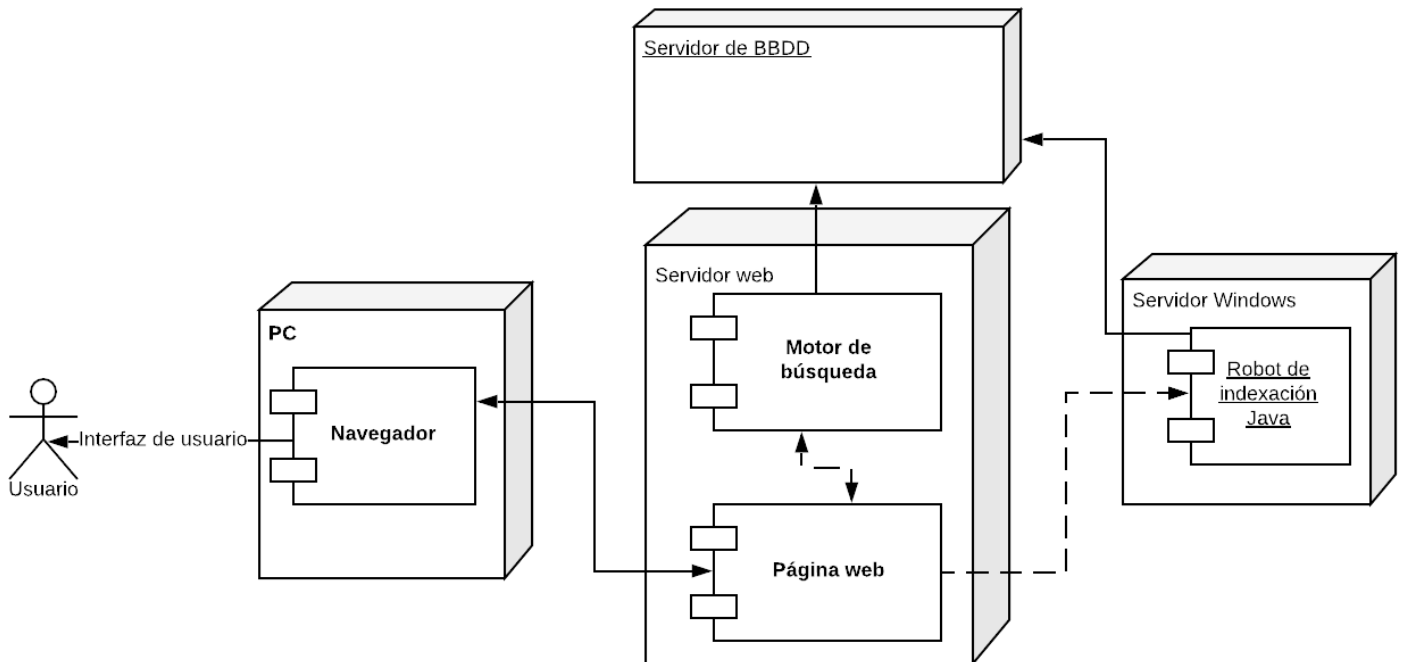
<b>Caso de uso: Administrar sitio</b>
<b>ID: CU4</b>
<b>Descripción:</b> El usuario administrador realiza tareas de administración del sitio
<b>Actores principales:</b> Usuario
<b>Actores secundarios:</b> BBDD
<b>Flujo principal:</b> <ol style="list-style-type: none"> <li>1. El caso de uso empieza cuando el administrador entra al panel de administración</li> <li>2. En él puede: <ol style="list-style-type: none"> <li>2.1 Consultar datos de la base de datos</li> <li>2.2 Optimizar base de datos</li> <li>2.3 Encender o detener el robot Java <ol style="list-style-type: none"> <li>2.3.1 Se envía una señal al Robot Java</li> </ol> </li> <li>2.4 Gestionar cuentas de usuarios</li> </ol> </li> </ol>

### 3.1.2. Diagramas de actividad

- *Funcionamiento del robot de indexación*

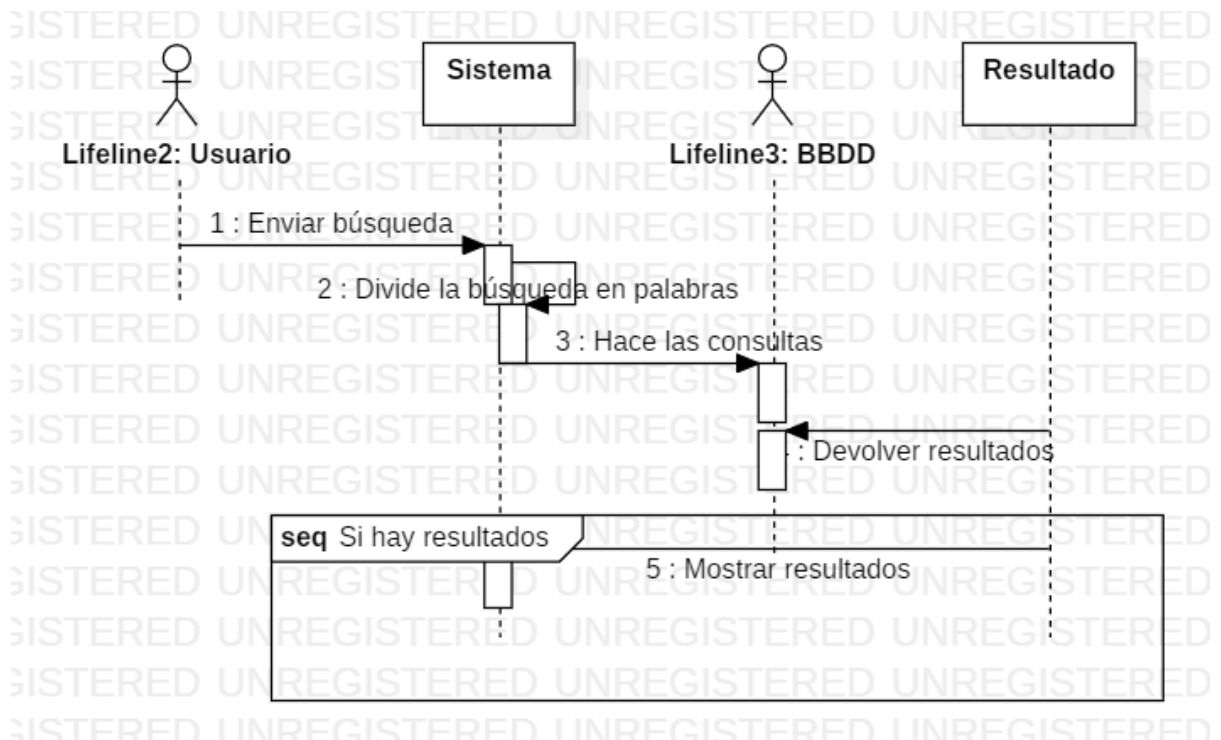


### 3.1.3. Diagrama de despliegue

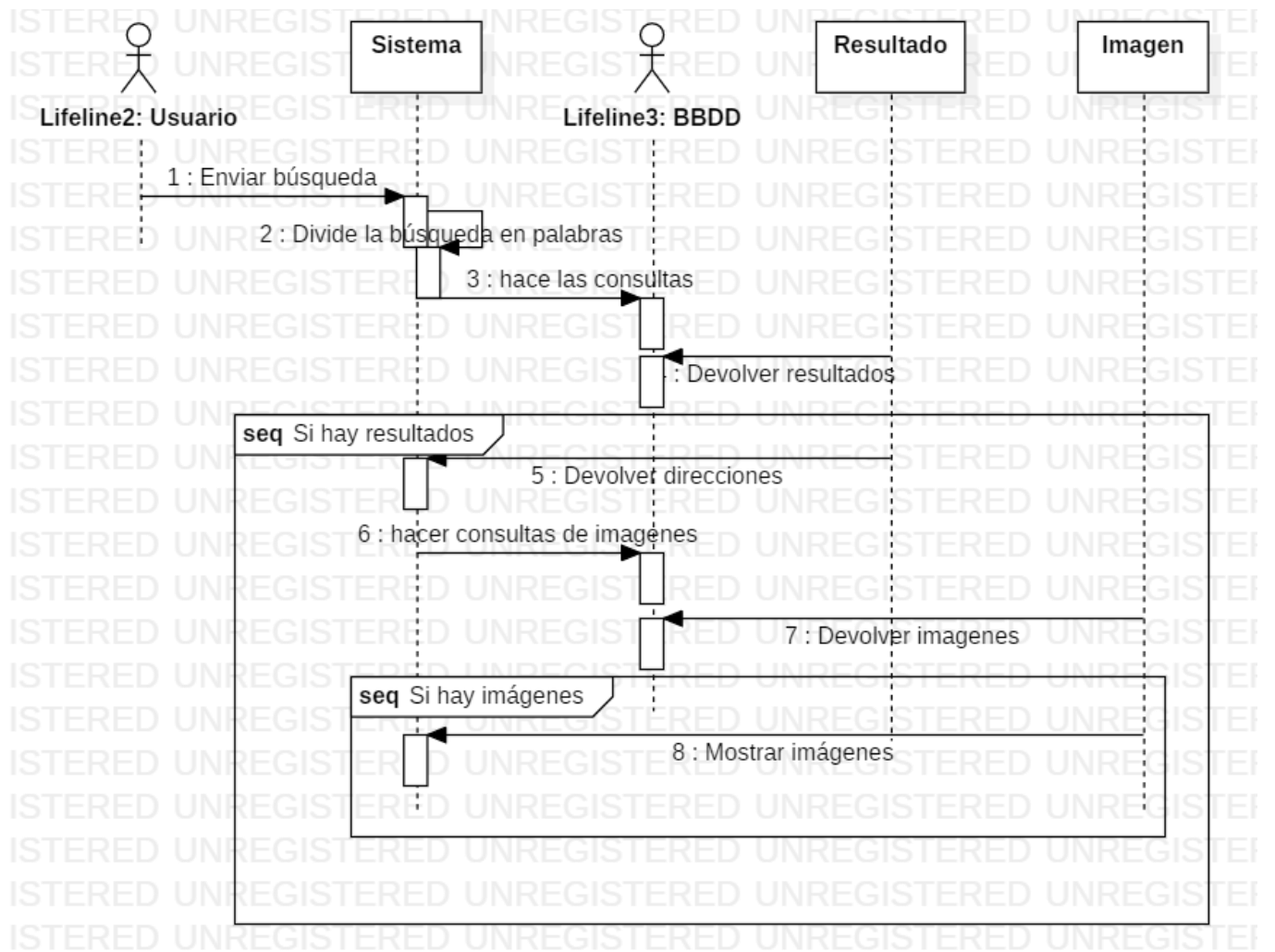


### 3.1.4. Diagramas de secuencia

- *Funcionamiento de la búsqueda de texto*

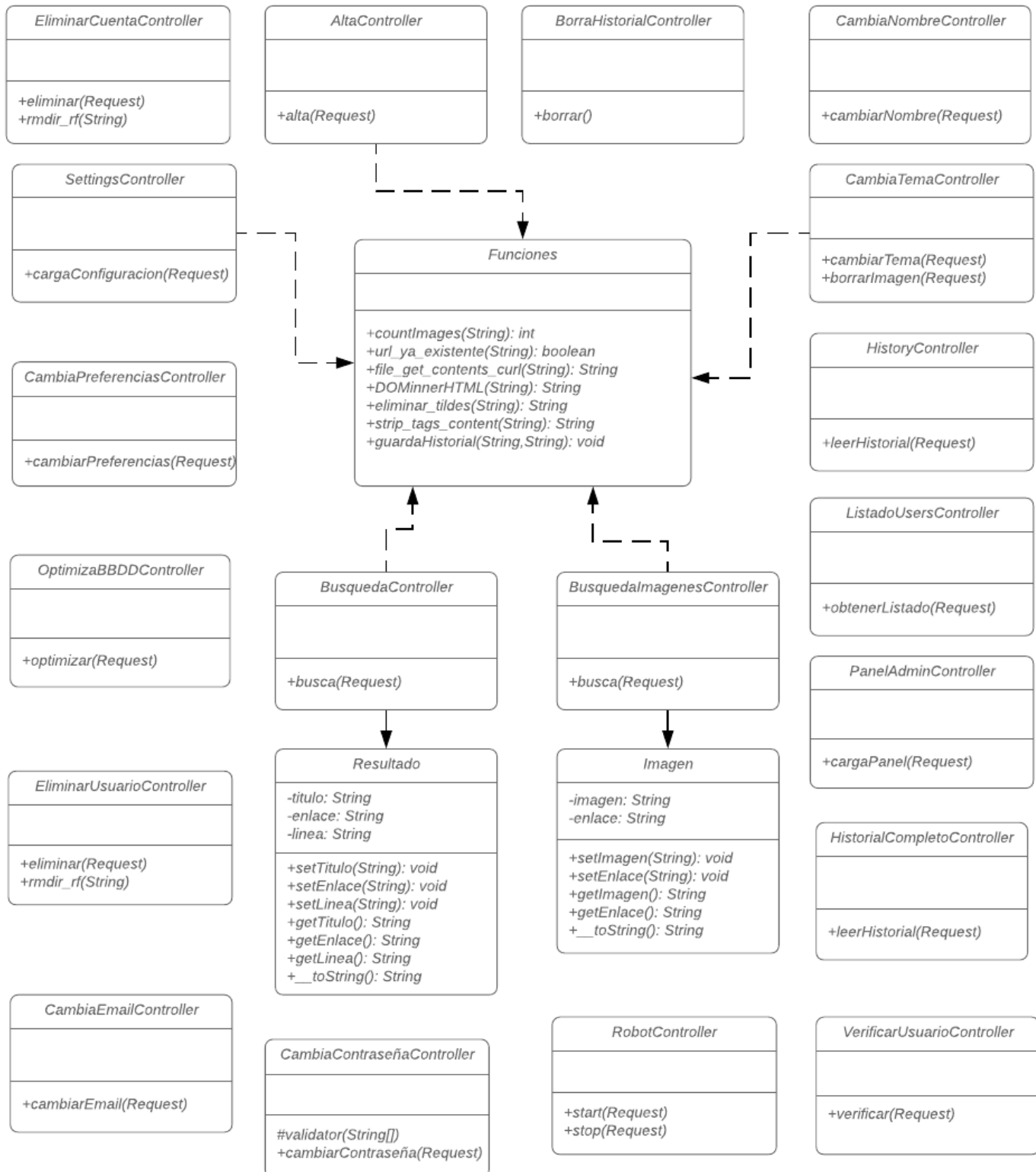


- *Funcionamiento de la búsqueda de imágenes*

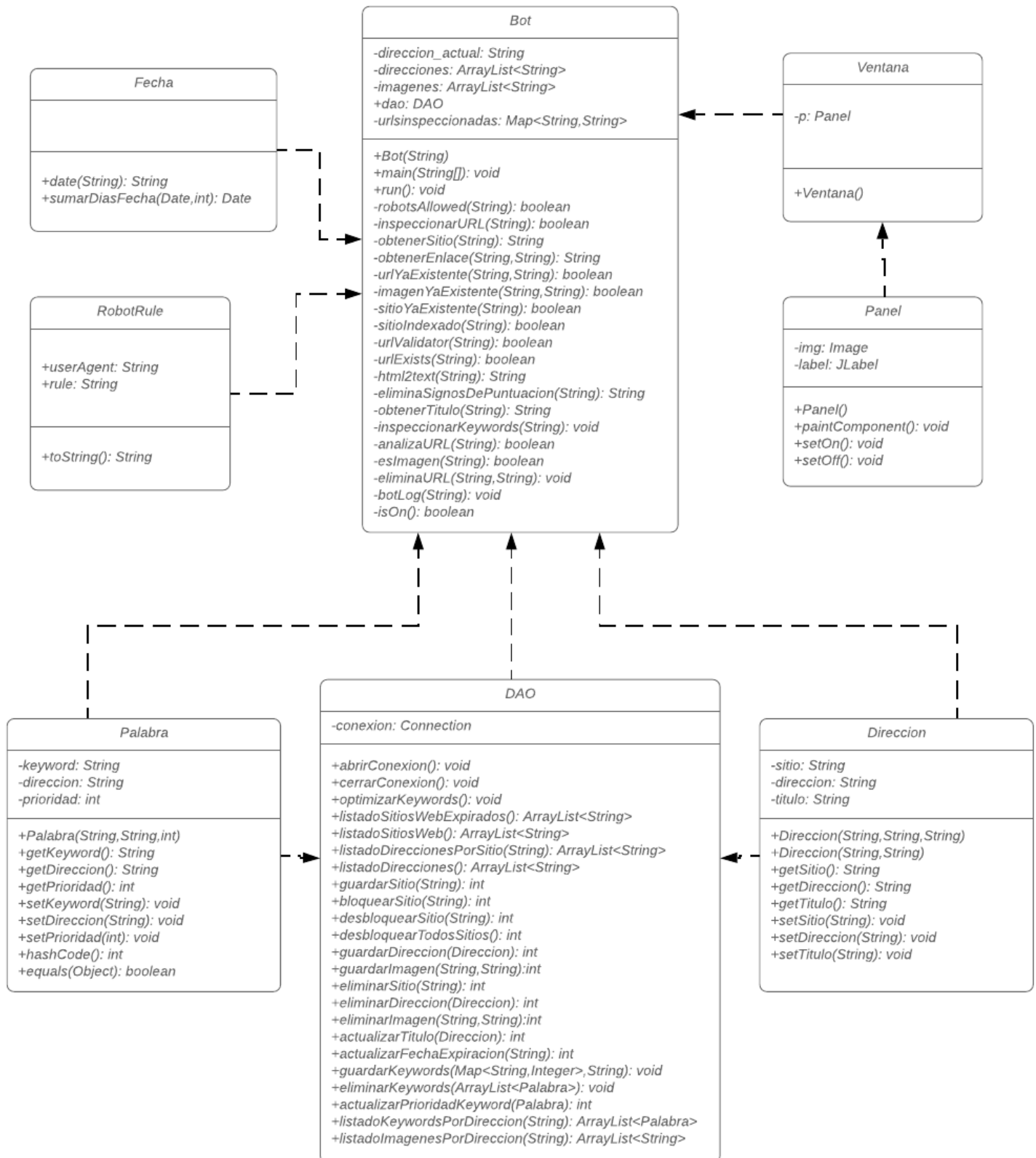


### 3.1.5. Diagramas de clases

- PHP



- Java



### 3.2. Estructura de la aplicación

La solución elegida para la estructura de la aplicación fue la de dividir la aplicación en dos lenguajes. La parte de la aplicación web, con el motor de búsqueda, ha sido desarrollada en PHP, y la parte del robot de indexación ha sido desarrollada en Java SE, para ejecutarse en el servidor de forma local e independiente de la aplicación web.

Se ha elegido esta solución porque permite la independencia entre las dos partes de la aplicación, y por lo tanto permite que los usuarios puedan visitar la web y hacer búsquedas con normalidad, aunque se esté ejecutando al mismo tiempo el robot.

Además, la aplicación web se desarrolló utilizando el framework de PHP **Laravel**, que utiliza el modelo MVC, y facilita el desarrollo al proporcionar muchas funciones ya hechas.

### 3.3. Funcionamiento Robot - Motor de búsqueda

Para el funcionamiento del robot y el motor de búsqueda se ha elegido la solución de indexar tanto las direcciones, como las palabras de cada página web, ya que la velocidad del robot no es tan importante como la del motor de búsqueda, porque la primera no afecta a la experiencia del usuario y la segunda sí, en gran medida. De esta forma, las búsquedas tardan tan solo unos pocos segundos en realizarse.

Además, se ha programado de forma que 20 robots pueden ejecutarse al mismo tiempo, utilizando Threads de Java. Para solucionar el problema de concurrencia que esto podría ocasionar, cuando un robot está inspeccionando un sitio lo bloquea con un campo de la base de datos.

El funcionamiento del robot, programado en Java, consiste en:

1. Obtener la lista de sitios almacenada en la base de datos que estén expirados y desbloqueados. Se considera “sitio” a la URL principal de cada página web.
2. Obtiene el primero de la lista, y analiza su URL con la API antivirus.
3. Si es segura, la bloquea en la base de datos (para evitar problemas de concurrencia), si no es segura la elimina de la base de datos.
4. Comprueba si la dirección ya existe en la tabla “direcciones”. En caso negativo, la añade.
5. Comprueba la dirección en el fichero robots.txt con una API. Si no está permitida, la elimina y vuelve al paso 1, si está permitida entra a inspeccionarla.
6. Para inspeccionarla, ejecuta el comando cURL (que debe estar instalado en el sistema). Este comando permite hacer peticiones a cualquier URL, establecer el “User Agent”, acceder sin tener en cuenta la caché, seguir redirecciones y pasar parámetros, entre otras opciones; y devuelve la respuesta de la petición (el contenido de la página web).



7. El robot analiza el contenido, busca enlaces, imágenes, y palabras, pasando por filtros que eliminan etiquetas html, signos de puntuación, caracteres extraños, evitan scripts y hojas de estilos y en el caso de las imágenes valida que la dirección corresponda a una imagen real.
8. Guarda en la base de datos las palabras e imágenes, junto a la dirección correspondiente, haciendo un índice. Para las palabras, además, se guarda una prioridad. Esta prioridad consiste en el número de veces que aparece la palabra en esa dirección, más 10 si está en el título.
9. Los enlaces encontrados, si son externos al sitio que está siendo inspeccionado actualmente, guarda en la base de datos la URL principal para ponerlos en cola; y si son locales los guarda en la base de datos junto a su título y son inspeccionados individualmente siguiendo el mismo proceso (recursivamente), siempre que no hayan sido inspeccionados ya en esa vez.
10. Una vez que termina con todas las direcciones de un sitio, actualiza su fecha de expiración a dos días después, y lo desbloquea.
11. Vuelve al paso 1. Este proceso se ejecuta en un bucle infinito, de forma que continuamente se actualiza la base de datos.

Para mantener actualizado el registro de direcciones, keywords e imágenes, se hacen algunos pasos más que consisten en:

1. Obtener todas las direcciones de un sitio y las keywords e imágenes de cada dirección en tres arrays.
2. Por cada dirección, keyword o imagen encontrada, se elimina del respectivo array si la contiene.
3. Si al final alguno de los arrays no está vacío, significa que esos registros ya no existen realmente, y por lo tanto se eliminan.

El funcionamiento del motor de búsqueda, programado en PHP, consiste en:

1. El usuario escribe en el buscador la palabra o palabras que quiere buscar y hace submit del formulario.
2. El motor de búsqueda recibe por POST la búsqueda del usuario. Divide la búsqueda por palabras y hace tantas consultas como numero de palabras contenga la búsqueda, y una con todas las palabras juntas. Por ejemplo, en una búsqueda de “hola mundo”, buscaría las direcciones que contengan “hola”, las direcciones que contengan “mundo”, y las direcciones que contengan tanto “hola” como “mundo”.
3. Devuelve las direcciones junto al título de cada una y las muestra al usuario, ordenadas por prioridad de mayor a menor (para devolver resultados más precisos), y paginadas con 10 resultados por página.



Con este método se ha conseguido que las búsquedas sean muy rápidas, tardando muy pocos segundos. Pero para ello se han tenido que “rechazar” algunas opciones que mejoran la calidad de búsqueda. No obstante, están disponibles para que el usuario las active en checkbox si lo prefiere. Estas opciones son:

- Búsqueda detallada: Esta opción consiste en hacer las consultas SQL utilizando la comparación “LIKE” en lugar de “=”, de forma que busca no solo palabras exactas, si no también parecidas, y abarca más resultados y puede ser útil para el usuario si no pone una palabra completa en el buscador.
- Vistas previas: Las vistas previas consiste en mostrar debajo de cada resultado un párrafo de la página web donde contiene la/s palabra/s buscadas. Para hacer esto, el motor de búsqueda tiene que entrar a cada dirección encontrada mediante el comando cURL ejecutado desde PHP, buscar ese párrafo y mostrarlo. A pesar de que esto reduce la velocidad de búsqueda, solo son unos pocos segundos, porque solamente entra a las 10 direcciones de la página que se está mostrando.

### 3.4. Búsqueda de imágenes

Las imágenes son indexadas de la misma forma que las palabras, referenciadas a una dirección. De esta forma, cuando el motor de búsqueda hace una búsqueda de imágenes, sigue un procedimiento parecido al de texto:

1. El usuario escribe en el buscador la palabra o palabras que quiere buscar y hace submit del formulario.
2. El motor de búsqueda recibe por POST la búsqueda del usuario. Divide la búsqueda por palabras y hace tantas consultas como numero de palabras contenga la búsqueda, y una con todas las palabras juntas.
3. Una vez obtenida la lista de direcciones, realiza otra consulta para obtener todas las imágenes de cada una de las direcciones.
4. Devuelve el resultado al navegador (paginado por 12 imágenes en cada página).

### 3.5. Robots.txt

Se ha utilizado la API crawlercommons 0.6 de Java. Esta API contiene muchas clases con las “reglas” que sigue la estructura de estos ficheros, y métodos que permiten consultar direcciones en los ficheros robots.txt de los sitios web.

### 3.6. Coordinación Robot - Motor de búsqueda

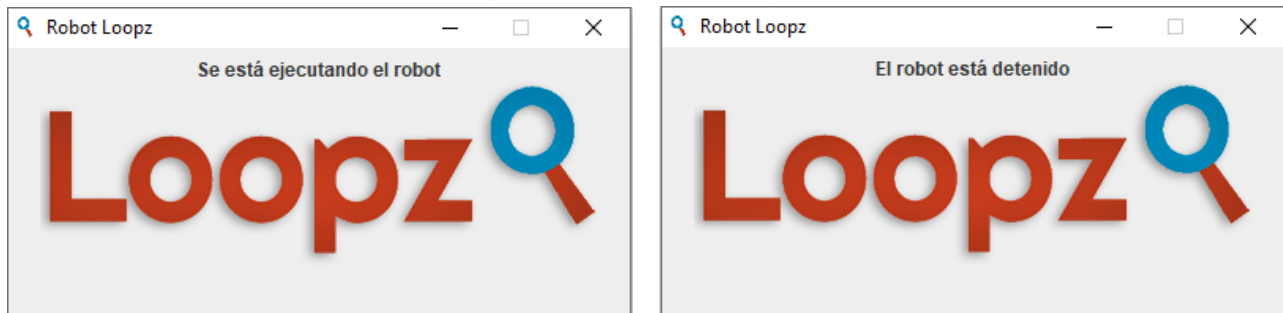
Para poder coordinar el funcionamiento del robot y el motor de búsqueda, se ha utilizado un fichero común a los dos ("robot\_status").

Consiste en un fichero alojado en el servidor web, cuyo contenido es únicamente 1 o 0 (1 encendido, 0 apagado). El programa Java continuamente comprueba el contenido de este fichero con una llamada cURL: si es 1 continúa su ejecución con normalidad, y si es 0 detiene todos los hilos que se encontraban ejecutándose y se queda en un bucle infinito en el cual continúa comprobando ese fichero, esperando a que vuelva a estar a 1 y reanudar su ejecución.

Desde la página web, gracias a una función que se ha añadido al panel de administración, se puede gestionar fácilmente el estado del robot (encender/apagar y comprobar estado actual) con un botón.



Al robot, que se ejecuta en un servidor Windows, se le ha implementado una interfaz gráfica diseñada con Java Swing. Esta interfaz contiene el botón de cerrar, para detener el programa fácilmente, una etiqueta que contiene el estado en el que se encuentra (ejecutándose o detenido), y el logotipo de Loopz en grande.



Se ha programado una función close() que se ejecuta al hacer click en el botón de cerrar, justo antes de cerrar el programa. Esta función ejecuta dos acciones importantes para evitar errores en la base de datos (como puede ser, que los sitios que se estaban inspeccionando se queden bloqueados permanentemente). Estas acciones son: desbloquear todos los sitios bloqueados por los Threads, y cerrar la conexión a la BBDD con la instancia de la clase DAO utilizada en la clase principal.

Para que este planteamiento funcione, el programa Java debe estar continuamente en ejecución, y su funcionamiento dependerá del fichero remoto “robot\_status”.

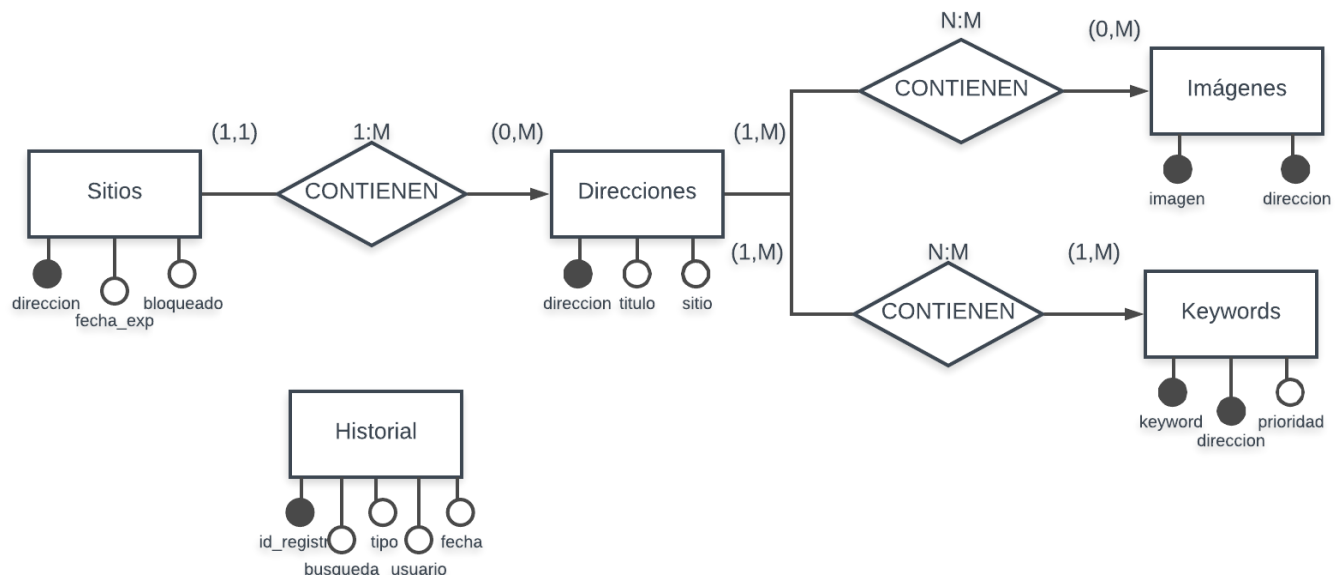
Por lo tanto, detener el programa desde el botón de cerrar, cerrará el proceso y hará que el programa no atienda al fichero. Esto solo debería hacerse en casos especiales que requieran un reinicio.

### 3.7. Sistema de almacenamiento de datos

Para el almacenamiento de datos se ha utilizado una base de datos relacional MySQL, y para algunos datos, demasiado simples para guardar en la BBDD, el sistema de ficheros.

#### 3.7.1. Base de datos

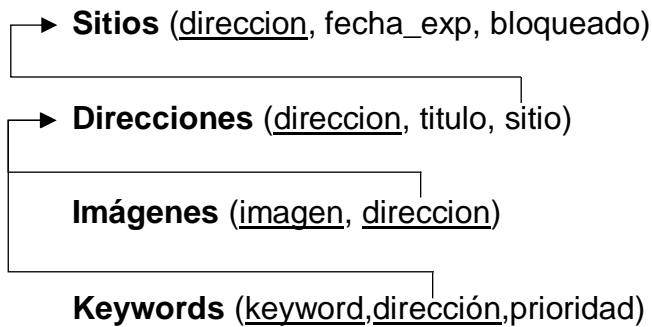
- *Modelo Entidad-Relación (MER)*



- **Sitios**: Almacena los sitios web, para ser analizados por el Robot Java.
  - *Dirección*: Es la URL principal de cada sitio web.
  - *Fecha\_exp*: Es la fecha que indica la expiración de la última indexación de este sitio. Una vez pasada, el sitio se pondrá a la cola para ser actualizado por el Robot Java.
  - *Bloqueado*: Indica si el sitio está bloqueado por el Robot. Se utiliza para evitar concurrencia entre varios Threads.

- **Direcciones:** Contiene las direcciones de los diferentes sitios web, y son mostradas en los resultados de búsquedas.
  - *Dirección:* Es la URL de cada registro.
  - *Título:* Es el título de esta URL (el contenido de la etiqueta <title>)
  - *Sitio:* Indica la URL del sitio al que pertenece cada dirección.
- **Imágenes:** Contiene las imágenes de las diferentes direcciones, y son mostradas en los resultados de imágenes.
  - *Imagen:* Es la URL que corresponde a cada imagen.
  - *Dirección:* Es la dirección a la que pertenece cada imagen.
- **Keywords:** Contiene las palabras de las diferentes direcciones, y se utilizan como índice para obtener las direcciones en las búsquedas.
  - *Keyword:* Es la palabra de cada registro.
  - *Dirección:* Es la dirección a la que pertenece cada palabra.
  - *Prioridad:* Es la prioridad de clasificación de cada palabra. Se utiliza para ordenar los resultados de las búsquedas.
- **Historial:** Almacena todas las búsquedas hechas en la página web.
  - *Id\_registro:* Es el identificador de cada registro.
  - *Búsqueda:* Es el contenido de cada búsqueda.
  - *Tipo:* Es “texto” o “imagen”, para diferenciar el tipo de búsqueda.
  - *Usuario:* Es la id del usuario que realizó la búsqueda. Cuando un usuario borra su historial, este valor se actualiza a 0. Las búsquedas por usuarios no registrados también corresponden al valor 0.
  - *Fecha:* Es la fecha en la que se hizo la búsqueda.
- **Relación Sitios-Direcciones:** Un sitio puede contener ninguna o muchas direcciones, pero una dirección solo puede pertenecer a un sitio (1:M).
- **Relación Direcciones-Keywords:** Una dirección puede contener una o muchas keywords, y una keyword puede pertenecer a uno o a muchos sitios. (N:M)
- **Relación Direcciones-Imágenes:** Una dirección puede contener ninguna o muchas imágenes, y una imagen puede pertenecer a uno o a muchos sitios. (N:M)

## • Grafo Relacional



**Historial** (id\_registro, búsqueda, tipo, usuario, fecha)

A mayores, existen tres tablas más, generadas por Laravel automáticamente en una migración, que son utilizadas para el login y registro de la aplicación. Estas son:

**Users** (id, name, email, email\_verified\_at, password, remember\_token, created\_at, updated\_at)

**Password\_resets** (email, token, created\_at)

**Migrations** (id, migration, batch)

## • Configuración

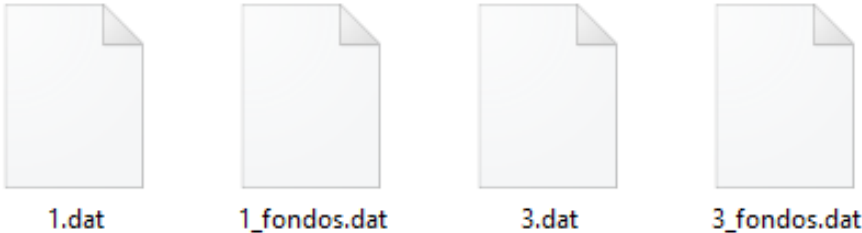
La tabla historial no se relaciona con la tabla users debido a que no se quiere que el campo “usuario” sea clave ajena del campo “id”. Esto es para que cuando un usuario borre el historial, en realidad no se borre, si no que se establezca el campo “usuario” a 0 (y esto no produzca error de integridad), y así tener un registro permanente de las búsquedas.

La base de datos debe estar optimizada para hacer consultas rápidamente, y para ello se ha creado un índice en la tabla keywords, que sirve para aumentar la velocidad de consultas. Además, la base de datos necesitará un mantenimiento, que consiste en ejecutar “OPTIMIZE TABLE keywords” periódicamente, para actualizar el índice existente. Esta tarea se puede realizar fácilmente desde el panel de administración.

Además, debe tener la codificación utf8mb4\_unicode\_ci para admitir todos los caracteres que se puedan encontrar en la red. Para tener esa codificación, la base de datos debe tener el formato “Barracuda” y además los campos de tipo VARCHAR solo deben tener 191 caracteres como máximo.

### 3.7.2. Ficheros

Los ficheros se han utilizado para almacenar las preferencias de los usuarios, que son tan solo tres valores SI/NO, por lo que cada usuario tiene un fichero con su ID que contiene tres números; y también para el sistema de fondos de pantalla de la página principal, que cada usuario puede configurar a su gusto, y estas preferencias se almacenan en un fichero que contiene los números de los fondos que muestra.



### 3.7.3. Imágenes

Las imágenes que los usuarios pueden subir para los fondos de pantalla se almacenan en carpetas dentro del servidor, con el id de usuario como nombre, y las imágenes numeradas del 1 al 5 (el máximo permitidas por usuario).

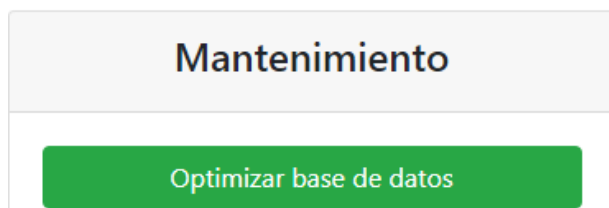


## 3.8. Mantenimiento

La base de datos, especialmente la tabla keywords, requiere de un mantenimiento debida al gran volumen de datos que almacena. Este mantenimiento consiste en ejecutar la sentencia SQL "OPTIMIZE TABLE keywords". Esto hará que MySQL vuelva a crear un índice sobre la tabla (que tendrá nuevos registros), con el fin de acelerar las consultas. Este mantenimiento es llevado a cabo de dos formas:

- **Automáticamente:** El programa del robot optimiza la tabla keywords automáticamente todos los días a las 12 a.m. y 12 p.m. utilizando la clase Calendar.

- **Manualmente:** Desde el panel de administración de la página web, se ha creado un botón que permite ejecutar esta sentencia SQL con un solo click. Estará a criterio del administrador ejecutar la optimización si las búsquedas se ralentizan demasiado.



### 3.9. Alta de sitios web

A pesar de no ser necesario, ya que el robot Java almacena nuevos sitios web cada vez que encuentra URLs externas al sitio que está inspeccionando, se ha implementado la posibilidad de que los usuarios registrados den de alta nuevos sitios web directamente en la base de datos, para que puedan registrar sitios más rápidamente, sin tener que esperar a que el robot los encuentre en la red.

Para ello, se ha diseñado un formulario en la página web, de forma que cuando un usuario envía una dirección, se extrae de ella la dirección principal ("sitio") y se comprueba si ya existe en la tabla sitios de la base de datos. En caso negativo, se inserta en la tabla y así está a la cola para que el programa Java las inspeccione.

### 3.10. Seguridad

#### 3.10.1. Página web

Las medidas básicas de seguridad en la página web son proporcionadas por el framework Laravel, como la validación de datos, prevención de inyección por formularios, el acceso a URLs prohibidas...Y se utilizan alerts de Bootstrap para mostrar mensajes de error o de éxito.

También se presentó otra brecha de seguridad importante con la subida de imágenes de los usuarios para el carrusel de la página principal, ya que las rutas de las imágenes eran `users/{id del usuario}/{número de imagen}.jpg` y esto permitía que cualquier usuario pudiera mostrar imágenes de otro usuario tan solo cambiando la ruta desde el navegador.

Para solventar este problema, se ha hecho que la imagen sea mostrada desde código PHP, de forma que la ruta mostrada al usuario simplemente es `imagen/{número de imagen}`, y desde el código se obtiene la id del usuario (encapsulándola) y la imagen correspondiente. Además, se movió la carpeta de imágenes fuera de la carpeta public. De esta forma los usuarios no tienen forma

de acceder a las imágenes de otros, porque la id queda encapsulada, y la carpeta de imágenes es inaccesible para ellos.

Para conseguir esto, se creó la ruta en el fichero web.php y se relacionó a una función, que valida si la imagen existe y la muestra si es así.

```
Route::get("/imagen/{num}",function($num){
    $id=Auth::id();
    $rutaImagen=base_path()."/users/images/$id/$num.jpg";
    if(file_exists($rutaImagen)){
        $informacionImagen = getimagesize($rutaImagen);
        header("Content-type: {$informacionImagen['mime']}");
        readfile($rutaImagen);
    }
});
```

### 3.10.2. *Robot de indexación Java*

En cuanto al problema de seguridad en el robot de indexación, en Java, ha sido solventado utilizando la API VirusTotal, que tiene funciones para escanear URLs fácilmente con una llamada cURL que devuelve los resultados de análisis de 66 antivirus.

Así, si la API devuelve algún positivo, el robot evita esa URL, y queda solventado el riesgo de acceder a páginas maliciosas.

Sin embargo, la versión gratuita solo permite 4 análisis por minuto, así que solo ha sido posible que el robot analice 4 de todas las direcciones a las que accede por minuto.



### 3.10.3. Cifrado HTTPS

Para garantizar la seguridad de sus datos a los usuarios, mediante la encriptación de sus datos en la conexión con la página web, se ha implementado el cifrado HTTPS con un certificado SSL.

Para ello, se obtuvo un certificado de la autoridad certificadora (CA) Let's Encrypt, a través de la página web ZeroSSL que genera los códigos de certificado y clave privado para los dominios que se introduzcan, que fueron www.loopz.cf, loopz.cf, y mail.loopz.cf.

Este certificado se instaló en el servidor Skylium, a través del cPanel. Allí se substituyó por un certificado auto-firmado que se genera por defecto.

Domains	Issuer	Expiration	Description	FQDNs	Certificate Expiration	Document Root
<input type="radio"/> loopz.cf <input type="radio"/> mail. <input type="radio"/> loopz.cf <input type="radio"/> www. <input type="radio"/> loopz.cf	Self-Signed	5/12/20	Cert for "loopz.cf"	<input checked="" type="checkbox"/> loopz.cf <input checked="" type="checkbox"/> mail.loopz.cf <input checked="" type="checkbox"/> www.loopz.cf	8/13/19	/public_html
<input checked="" type="radio"/> loopz.cf <input checked="" type="radio"/> mail. <input checked="" type="radio"/> loopz.cf <input checked="" type="radio"/> www. <input checked="" type="radio"/> loopz.cf	Let's Encrypt	8/13/19	loopz.cf, mail.loopz.cf, and www.loopz.cf			

Por último, se configuró el fichero .htaccess de la aplicación para que utilice rutas https por defecto.

```

RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]

```

### 3.11. Registro de usuarios

Se ha implementado el registro de usuarios en la página web. Un usuario, al registrarse, tendrá algunos privilegios en la web:

- ❖ Dar de alta nuevos sitios web en el buscador.
- ❖ Almacenar su historial de búsqueda y poder borrarlo.
- ❖ Establecer sus preferencias de búsqueda.
- ❖ Modificar el diseño (tema) de la página.

El usuario con ID 1 es considerado el administrador, y tiene acceso al panel de administración.

### 3.12. Logs

La parte PHP de la aplicación contiene los ficheros log en la carpeta /storage/logs. En ella, diariamente, se crea un fichero log que almacena, con fecha y hora:

- ❖ Errores de la aplicación.
- ❖ Los inicios de sesión e intentos de inicio de sesión.
- ❖ Los registros de nuevos usuarios.
- ❖ Cada vez que se optimiza la base de datos manualmente.
- ❖ Cuando se eliminan cuentas de usuarios (por el propio usuario o por el administrador).
- ❖ Cambios en los datos de los usuarios (nombre, email o contraseña).

En cuanto a la parte de Java, existe un fichero bot.log en el servidor donde está el programa, en el cual se almacena, con fecha y hora:

- ❖ Cuando un robot entra en una URL .
- ❖ Cuando inspecciona una URL.
- ❖ Cuando actualiza una URL.
- ❖ Cuando elimina una URL.

### 3.13. Sistema de correo electrónico

Para evitar que los usuarios se registren con emails falsos, se ha activado la función de verificación de correo electrónico que ya venía incorporada, pero desactivada, en Laravel.

Para activarla, y que, además, la recuperación de contraseña por email funcione, ha sido necesario configurar en la página web un sistema de correo electrónico.

El hosting Skylium ofrece un servidor SMTP, y la posibilidad de crear cuentas de email con el dominio de la web. Por lo tanto, se aprovechó para crear una cuenta para el envío de los correos automáticos a los usuarios: [automatic@loopz.cf](mailto:automatic@loopz.cf).

Laravel también facilita la configuración a través del fichero .env, donde bastó con indicar el driver que se utilizaría (smtp), el host y puerto, una cuenta de email y contraseña, y la encriptación.

Para activar la verificación de correo electrónico, bastó con pasar un parámetro a las rutas de Auth: `Auth::routes(['verify' => true]);` y configurar las redirecciones después del registro y después de la verificación.

Además, se ha restringido el acceso a cualquier zona de la página web para usuarios que no hayan verificado su cuenta, añadiendo `->middleware('verified');` en las rutas de web.php, que redirige a la vista “verify” a los usuarios no registrados.

Además, se ha configurado el controlador “CambiaEmailController” para que, cuando un usuario cambie su email desde el panel de configuración, también tenga que verificar su email.

También se han creado los correos [info@loopz.cf](mailto:info@loopz.cf) y [daniel.lopez@loopz.cf](mailto:daniel.lopez@loopz.cf) para la página de contacto. El primero se utilizará para recibir correos de los usuarios sobre cualquier información que requieran, y el segundo será el correo del director, para temas más importantes.

Account ^ @ Domain	Restrictions	Storage: Used / Allocated / %
<input type="checkbox"/> > automatic@loopz.cf	✓ Unrestricted	56.05 KB / 1 GB / 0.01%
<input type="checkbox"/> > daniel.lopez@loopz.cf	✓ Unrestricted	0 bytes / 1 GB / 0%
<input type="checkbox"/> > info@loopz.cf	✓ Unrestricted	0 bytes / 1 GB / 0%

### 3.14. Estructura de la web

La página web se ha desarrollado con Laravel, utilizando el modelo MVC, de forma que toda la lógica (PHP) se programa en controladores y se separa de las vistas (HTML, CSS, JS, Blade).

El fichero de rutas web.php contiene las rutas de la aplicación y a que se dirigen (a vista o a controlador), y el acceso a la base de datos se ha hecho con las facilidades que ofrece Laravel con el fichero .env.

Para las consultas del motor de búsqueda se utilizó el Query Builder, salvo alguna consulta demasiado compleja que se escribió directamente en SQL; y para el login se utilizó Eloquent y el modelo User.

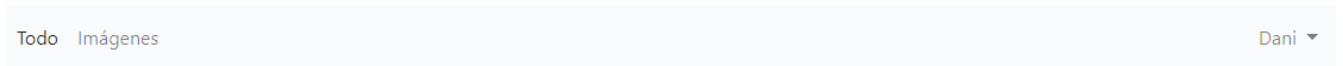
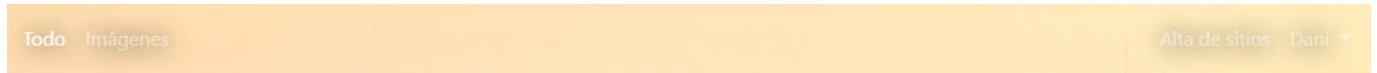
El diseño de la interfaz web se hizo con el sistema de layouts de Laravel, de forma que hay tres layouts principales y el resto de vistas heredan de ellos. Además, se utilizó la librería Bootstrap para el diseño y la librería JQuery para la lógica en el entorno del cliente.

Para el panel de administración se utilizó un gráfico de la librería de PHP FusionCharts. La interfaz es completamente responsive, adaptable a todos los tamaños de pantalla.

### 3.14.1. Cabeceras

Se diseñaron tres cabeceras similares, que corresponden a los tres layouts:

- “Main” se utiliza en las páginas principales (la de búsqueda y la de búsqueda imágenes) y los formularios de login y registro. Muestra el carrusel de fondo de pantalla (si está activada su visualización)



En pantallas pequeñas el menú se hace desplegable y se posiciona a la izquierda.



- “Secondary” se utiliza en el resto de las vistas excepto en las de resultados. Contiene el logo de Loopz a la izquierda como enlace a la página principal.



En pantallas pequeñas el menú se hace desplegable y se posiciona a la derecha. A la izquierda se mantiene el logo de Loopz.



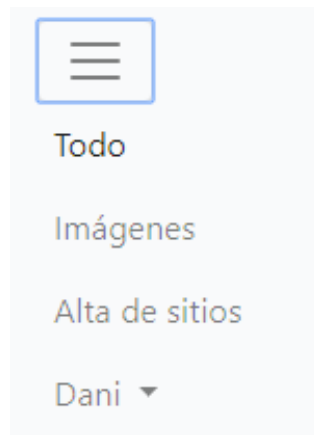
- “Search” se utiliza en las páginas de resultados. Contiene un mini buscador a la izquierda, para poder volver a buscar desde la propia página de resultados.



En pantallas pequeñas el menú se hace desplegable y se posiciona a la derecha y a la izquierda se mantiene el buscador.

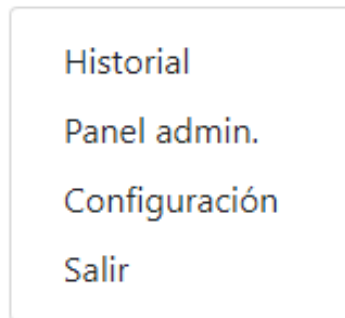


El menú desplegable de las pantallas pequeñas, muestra los enlaces verticalmente:



En todos los layouts hay una parte común, que son los enlaces “Todo” e “Imágenes” que corresponden a la página principal y a la página de búsqueda de imágenes, respectivamente; los enlaces de “Iniciar sesión” y “Registrarse” (para no registrados) y el de “Alta de sitios” y el menú desplegable (para registrados), que contiene las opciones de Historial, Configuración, Panel de Admin. (exclusivamente el usuario administrador), y Cerrar sesión.

Alta de sitios Dani ▼



En el layout “search” los enlaces “Todo” e “Imágenes” no redirigen a las páginas principales, sino a las páginas de resultados de la búsqueda que se haya hecho.

### 3.14.2. *Pie de página*

En todas las páginas de Loopz aparece al final un pie de página con enlaces informativos sobre la web. Estos son “Política de cookies”, “Condiciones” y “Acerca de”.

En la página principal su posición varía según el tamaño de pantalla. En pantallas grandes aparece a la izquierda de los indicadores del carrusel (en caso de estar este activado), y en pantallas pequeñas los indicadores pasan a estar por encima del pie de página.



En las páginas que utilizan los layouts search y secondary, aparece el pie de página centrado en todos los tamaños de pantalla.

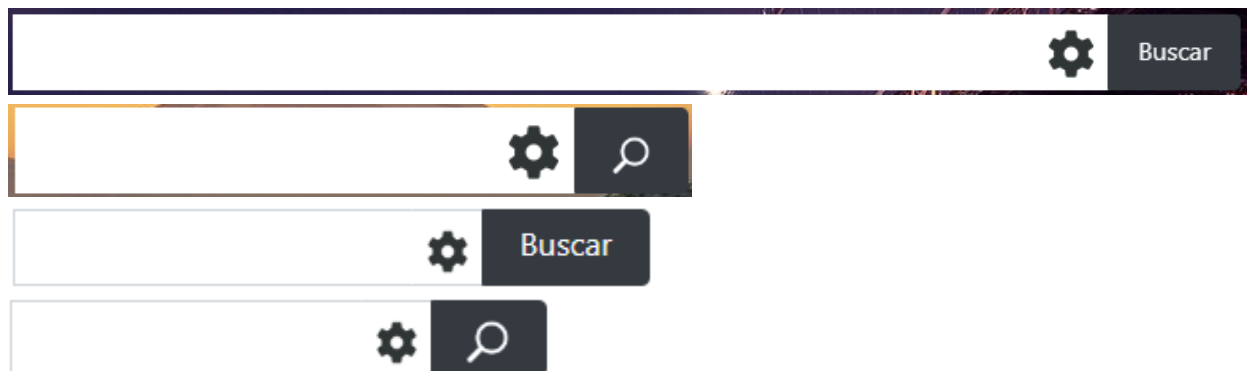
[Política de cookies](#) [Condiciones](#) [Acerca de](#)

### 3.14.3. *Buscador*

El buscador aparece en las páginas principales y en las de resultados (en estas, en el layout “search”).

Es un formulario horizontal, con el campo de texto a la izquierda y el botón submit a la derecha, integrado en el campo de texto utilizando los input-groups de Bootstrap. En las páginas principales aparece en tamaño grande (lg), y en las de resultados en tamaño normal.

El botón submit contiene la etiqueta “Buscar”, y en pantallas pequeñas se cambia por un icono de una lupa para hacer más pequeño el botón y mantener más grande el campo de búsqueda.

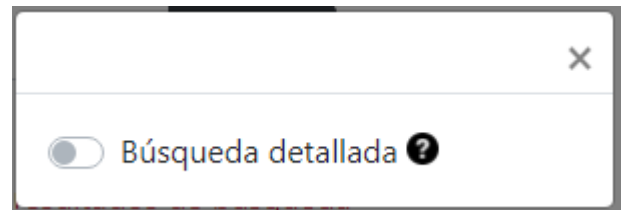
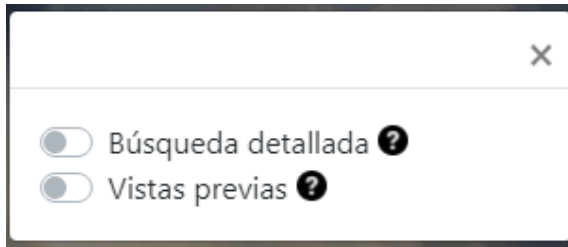


Al hacer click en el botón submit, se cambia la etiqueta o icono por un “spinner” para indicar visualmente que se está realizando la búsqueda.

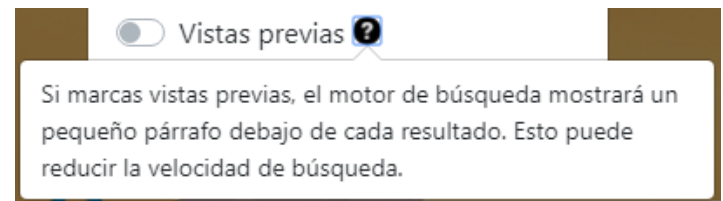
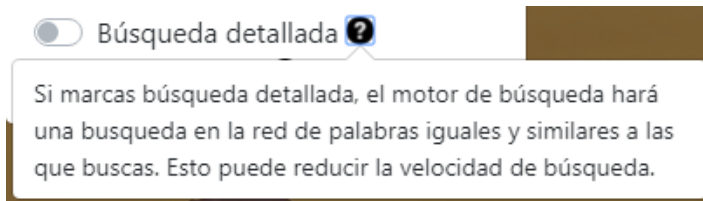


También tiene un botón configuración el cual, al pulsarlo, abre una ventana en la que se pueden activar la “búsqueda detallada” y las “vistas previas” (estas solo disponibles en la búsqueda de texto).

Para usuarios no registrados, se guardarán sus preferencias en cookies con caducidad de 1 hora; y los usuarios registrados pueden guardar sus preferencias con tiempo ilimitado en su panel de configuración.

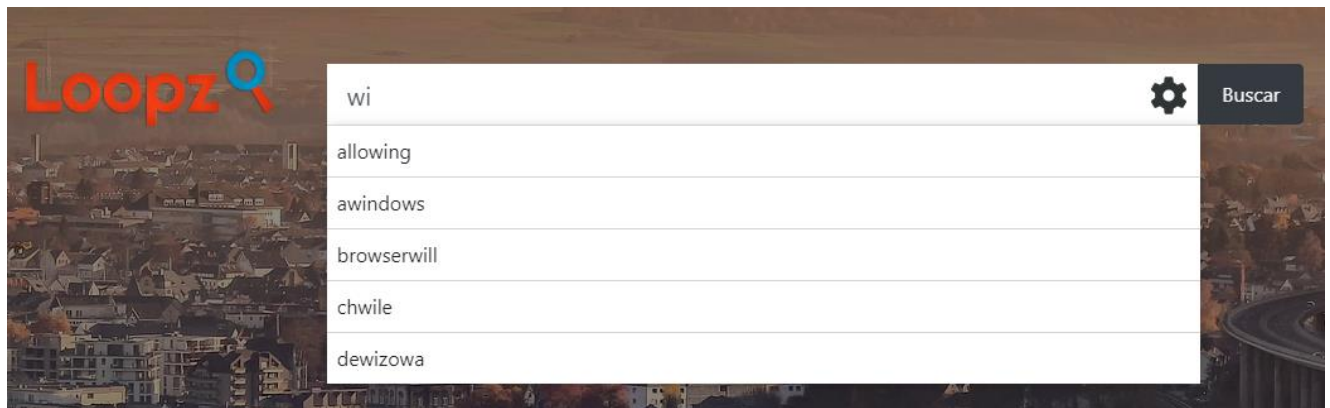


Los botones (?) muestran popovers que explican en qué consisten estas opciones.



También se le ha implementado al buscador la función de autocompletar que, mediante una función AJAX, muestra sugerencias de completado de búsqueda en base a las palabras registradas en la base de datos.

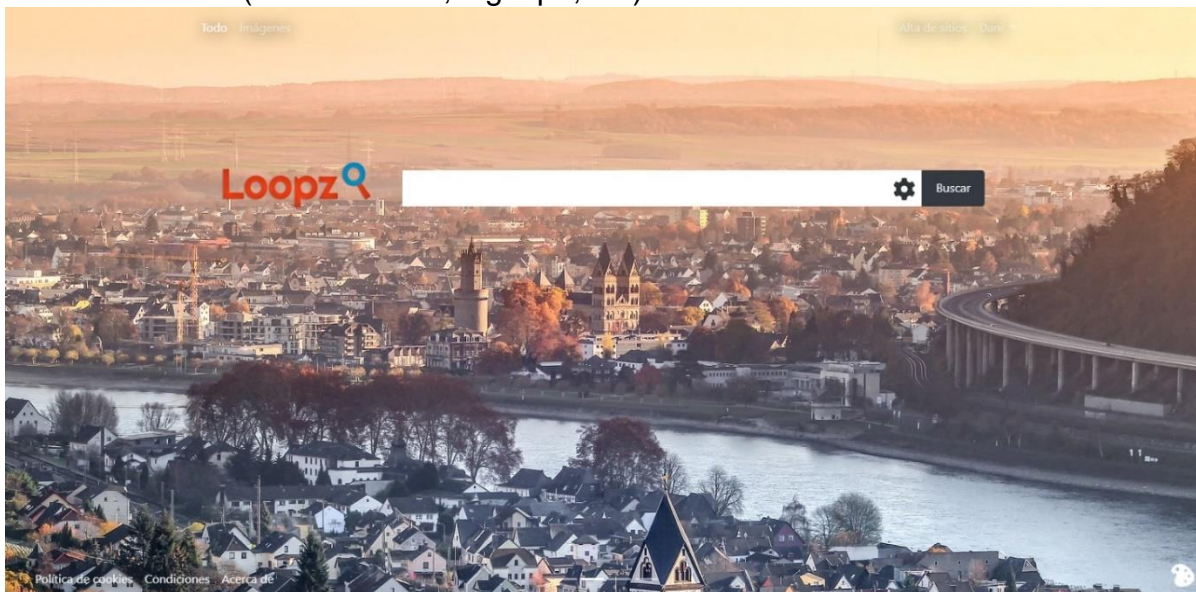
Funciona tanto en la página principal como en la de resultados.



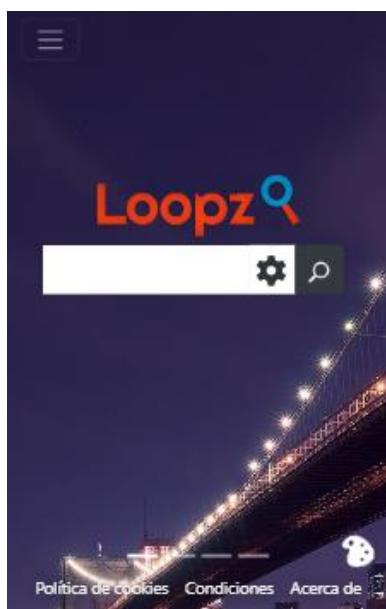


### 3.14.4. Páginas principales

Las páginas principales se han diseñado en una vista común llamada “home”. Es la que contiene el buscador en grande con el logotipo de Loopz. Se utiliza tanto en búsqueda de texto como en búsqueda de imágenes, con algunas diferencias entre ambas (como el título, logotipo, etc).

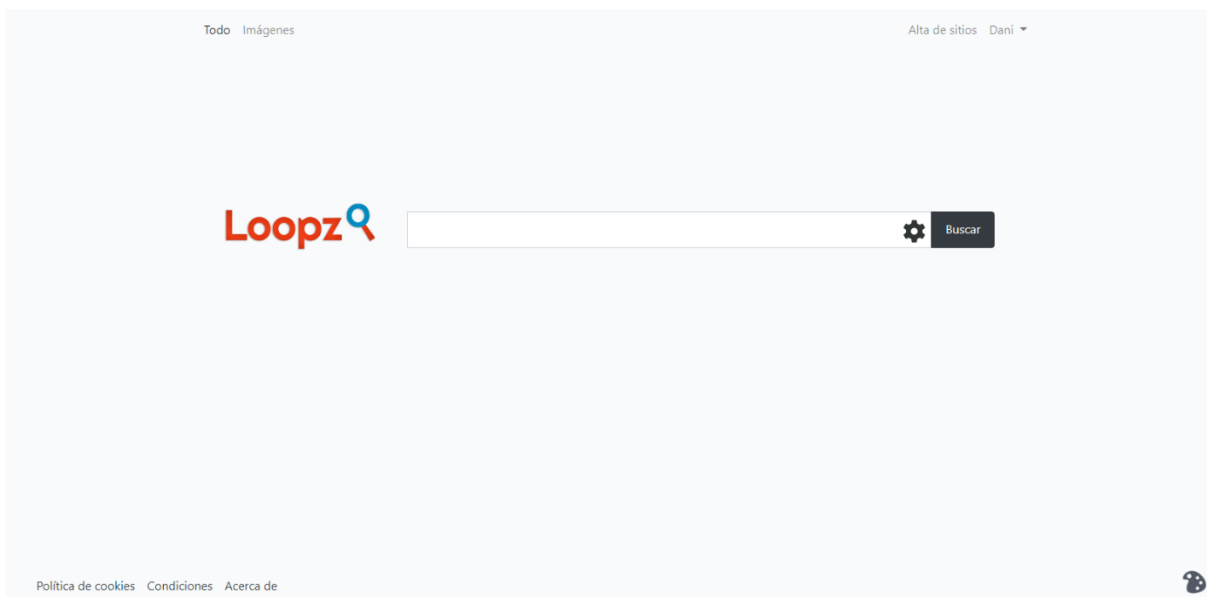


Tiene la particularidad de tener un carrusel de imágenes como fondo de pantalla. Este carrusel puede ser modificado por los usuarios registrados a su gusto, subiendo nuevos fondos o seleccionando cuales quiere que se muestren. En pantallas pequeñas, el logo pasa a estar encima del buscador y centrado, y el buscador se adapta de la misma forma que el layout “search”, para que el botón ocupe menos espacio, y el campo de búsqueda más.





También es posible la opción de desactivar el carrusel, que se puede hacer desde la propia página principal con el botón que se encuentra abajo a la derecha (visible solo por usuarios registrados), o desde configuración.



Al desactivarlo, el buscador se posiciona más al centro de la pantalla, y la cabecera pasa a tener texto color negro.

En la página de imágenes, el logotipo de Loopz varía un poco, ya que se le añade un subtítulo "Imágenes":



### 3.14.5. Páginas de resultados

La página de resultados es la que se muestra después de la búsqueda de texto o imágenes. En el caso de texto, contiene una lista vertical de resultados, cada uno de ellos con un título, debajo la URL, y debajo, en caso de estar activadas las vistas previas, un párrafo de la página web.

Al final aparecen los botones para moverse entre las páginas.



Se han encontrado 21 resultados

[Wikipedia – Die freie Enzyklopädie](#)  
<https://de.wikipedia.org/>  
**wikipedia:**Hauptseite aus **wikipedia**, der freien Enzyklopädie Zur Navigation springen Zur Suche springen Willkommen bei **wikipedia** **wikipedia** ist ein Projekt zum Aufbau einer Enzyklopädie aus freien Inhalten, zu denen du sehr gern beitragen kannst. Seit Mai 2001 sind 2.301.047 Artikel ...

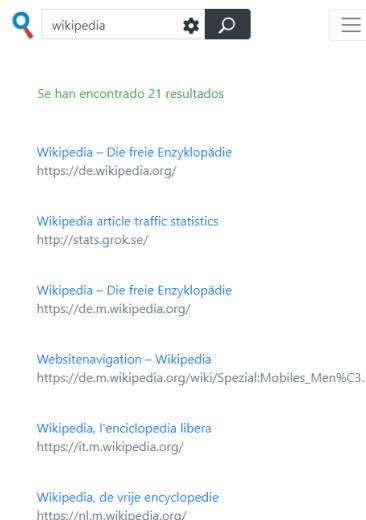
[Wikipedia article traffic statistics](#)  
<http://stats.grok.se/>  
**wikipedia** article traffic statistics What do **wikipedia**'s readers care about? Is Britney Spears more popular than Brittany? Is Asia Carrera more popular than Asia? How many people looked at the article on Santa Claus in December? How many looked at the article on Ron Paul? ...

[MediaWiki](#)  
<https://mediawiki.org/>

1 2 3

Política de cookies Condiciones Acerca de

En pantallas pequeñas se adapta el texto de los títulos, enlaces y párrafos, para que no deforme la página en caso de superar el ancho. En ese caso, se corta con puntos suspensivos (utilizando propiedades de CSS3).



Se han encontrado 21 resultados

[Wikipedia – Die freie Enzyklopädie](#)  
<https://de.wikipedia.org/>

[Wikipedia article traffic statistics](#)  
<http://stats.grok.se/>

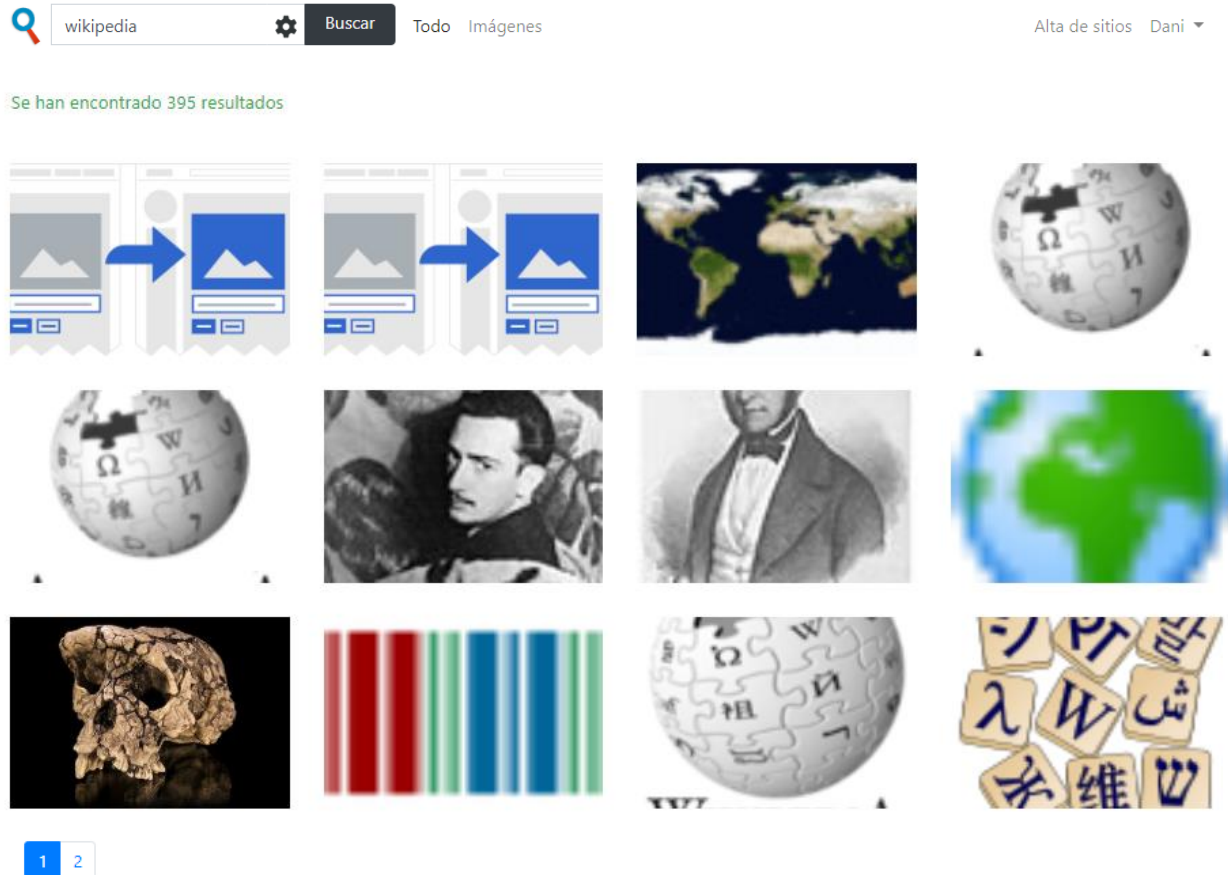
[Wikipedia – Die freie Enzyklopädie](#)  
<https://de.m.wikipedia.org/>

[Websitenavigation – Wikipedia](#)  
[https://de.m.wikipedia.org/wiki/Spezial:Mobiles\\_Men%C3...](https://de.m.wikipedia.org/wiki/Spezial:Mobiles_Men%C3...)

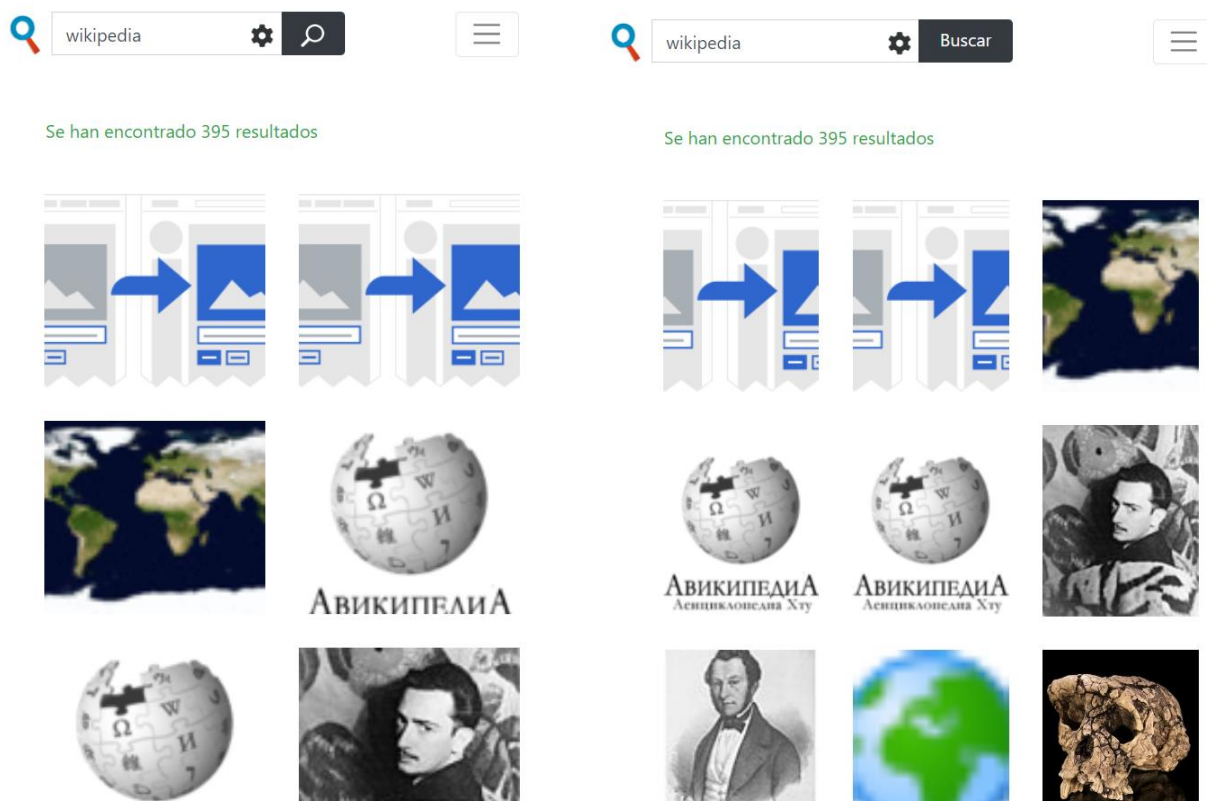
[Wikipedia, l'enciclopedia libera](#)  
<https://it.m.wikipedia.org/>

[Wikipedia, de vrije encyclopedie](#)  
<https://nl.m.wikipedia.org/>

En el caso de imágenes, se muestran 12 imágenes en horizontal, en una estructura cuadriculada:



En pantallas pequeñas se muestran menos columnas (3 en pantallas medianas y 2 en pantallas pequeñas), y las imágenes se adaptan, sin deformarse, a los tamaños correspondientes:



### 3.14.6. Página de alta de sitios

La página de alta de sitios web contiene título, descripción y formulario para dar de alta.

Loopz [Todo](#) [Imágenes](#)

[Alta de sitios](#) Dani ▼

#### Alta de sitios web en Loopz

El robot de Loopz rastrea cada URL del registro, guarda cada palabra encontrada y vuelve a rastrear cada enlace encontrado en un proceso recursivo.

De esta forma, todas las páginas web de la red son en algún momento rastreadas. Sin embargo, si una URL específica no aparece frecuentemente en otras páginas web, el proceso de encontrarla y, por lo tanto, rastrear ese sitio, puede llevar más tiempo.

Con este formulario, puedes hacer que el robot de Loopz acceda directamente a una URL antes de encontrarla en la red. Solamente debes introducir una URL cualquiera por cada sitio web, ya que el robot accede por medio de la URL principal.

Introduce una URL válida


[Dar de alta](#)

[Política de cookies](#) [Condiciones](#) [Acerca de](#)

### 3.14.7. Historial

La página del historial personal de cada usuario presenta una tabla con tres columnas: “Fecha”, “Búsqueda” y “Tipo”, en la cual aparece el historial de búsquedas, paginado por 20 registros en cada página.



Al final de la página aparecen los botones para cambiar de página y el enlace de Limpiar historial.


Todo Imágenes
Alta de sitios Dani ▾

Fecha	Búsqueda	Tipo
12:24 16/05/2019	wikipedia	imagen
12:24 16/05/2019	wikipedia	imagen
12:24 16/05/2019	wikipedia	texto
12:23 16/05/2019	wikipedia	imagen
10:14 16/05/2019	wikipedia	texto
10:10 16/05/2019	wikipedia	texto

1 2 3 5
Limpiar Historial

En pantallas pequeñas, la tabla reduce su ancho y el enlace de Limpiar historial y los botones de páginas pasan a ocupar todo el ancho y estar centrados.

Fecha	Búsqueda	Tipo
09:16 21/05/2019	wikipedia	texto
09:16 21/05/2019	wikipedia	texto
09:16 21/05/2019	wikipedia	texto
09:16 21/05/2019	wikipedia	texto
09:16 21/05/2019	wikipedia	texto
09:16 21/05/2019	wikipedia	texto
09:16 21/05/2019	wikipedia	texto

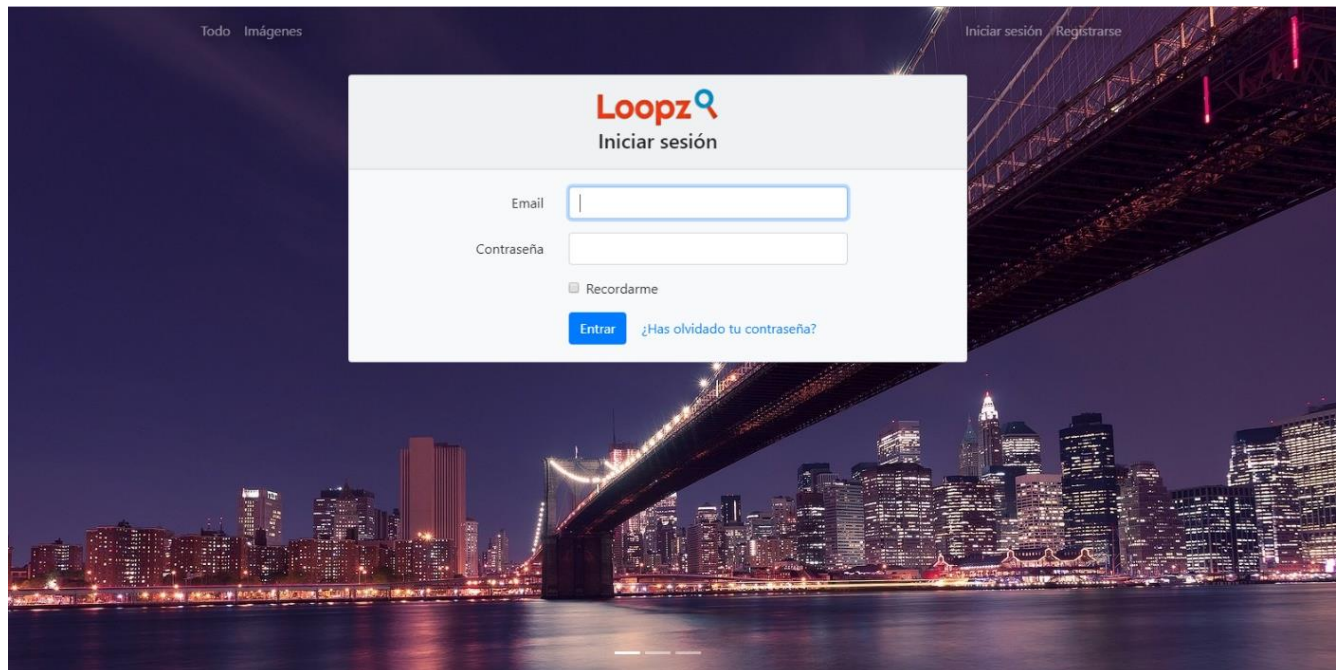
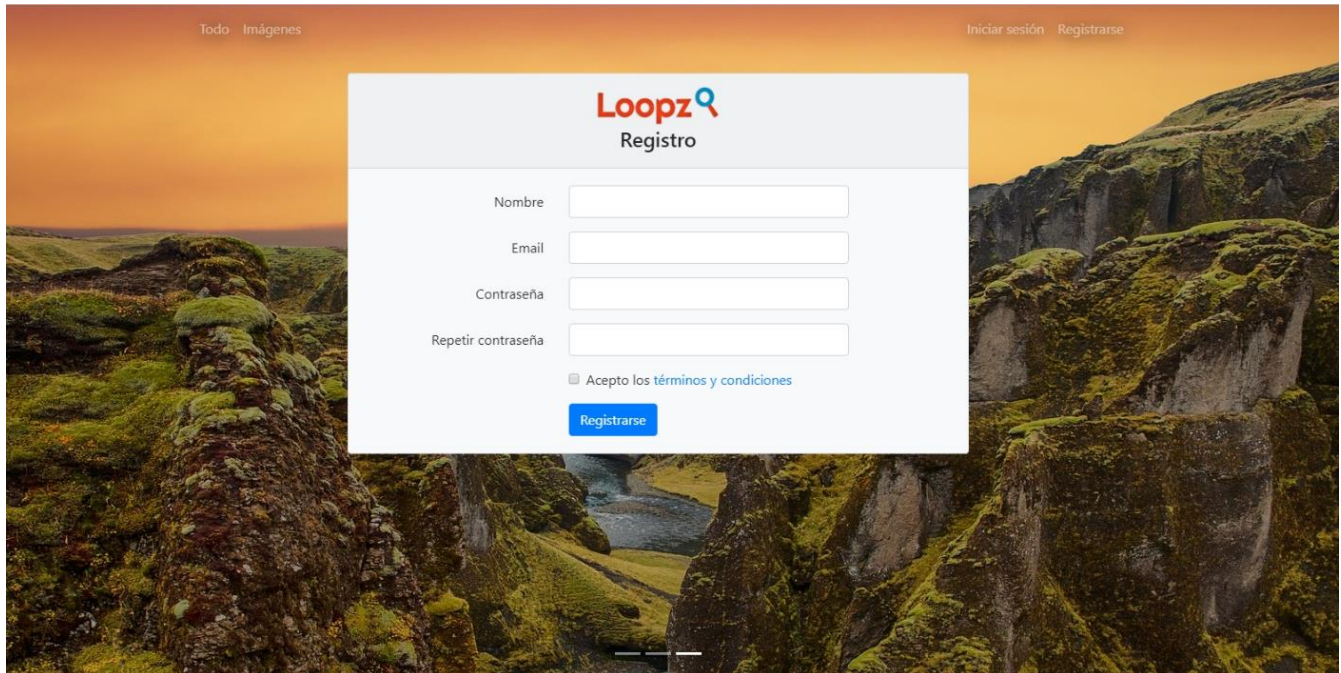
1
Limpiar Historial

Política de cookies
Condiciones
Acerca de



### 3.14.8. Login y Registro

Los formularios de login y registro se muestran con el layout “main”, por lo tanto muestran el carrusel de fondo. Utilizan las cards de Bootstrap, y en el header incluyen un h4 con el título y encima el logotipo de Loopz.



### 3.14.9. Verificación de email y recuperación de contraseña

Las páginas de recuperación de contraseña y verificación de email tienen el mismo diseño que las de login y registro, con el layout “main”, el carrusel de fondo, las cards de Bootstrap y el logotipo en el card-header.



Todo Imágenes

Iniciar sesión Registrarse

**Loopz**

Recuperación de contraseña

Email

Enviar link de recuperación



Todo Imágenes

Iniciar sesión Registrarse

**Loopz**

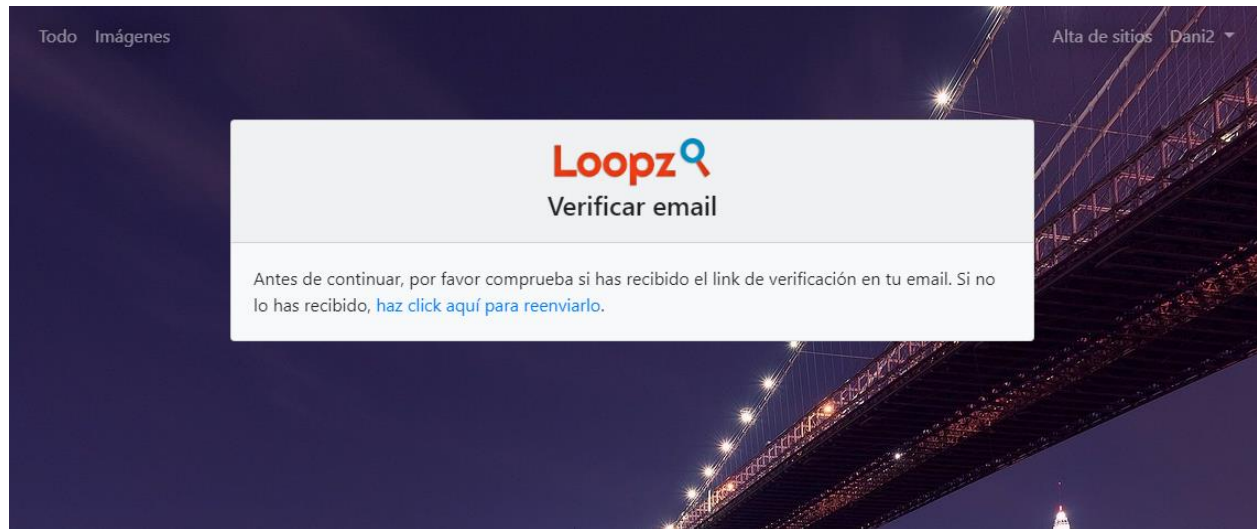
Recuperación de contraseña

Email

Contraseña

Confirm Password

Recuperar contraseña



En cuanto a los emails recibidos por los usuarios, también han sido modificados. El nombre del remitente es “Loopz” y el email remitente que se muestra es `no-reply@loopz.cf` para indicar al usuario que no responda.

En la cabecera aparece el logotipo de Loopz.

En el cuerpo muestran la descripción del email y el botón correspondiente, además de una URL por si el botón fallara en algún navegador.

En el pie de email se muestra el nombre de la empresa y el año: “Loopz S.A. 2019”.



¡Hola!

Estás recibiendo este email porque hemos recibido una petición de recuperación de la contraseña de tu cuenta de Loopz.

Recuperar contraseña

Este link de recuperación de contraseña expirará en 60 minutos.

Si no hiciste ninguna petición, ignora este mensaje.

Un saludo,  
Loopz S.A.

Si estás teniendo problemas haciendo click en el botón “Recuperar contraseña”, copia y pega esta dirección en tu navegador web: <https://loopz.cf/password/reset/58322d4054f2c1badf051c8afd1b5ac125339bdf21aa64a4b7720dc75cd7cdd1>

Loopz S.A. 2019



¡Hola!

Por favor haz click en el botón inferior para verificar tu email de tu cuenta de Loopz.

Verificar email

Si no has creado una cuenta, ignora este mensaje.

Un saludo,  
Loopz S.A.

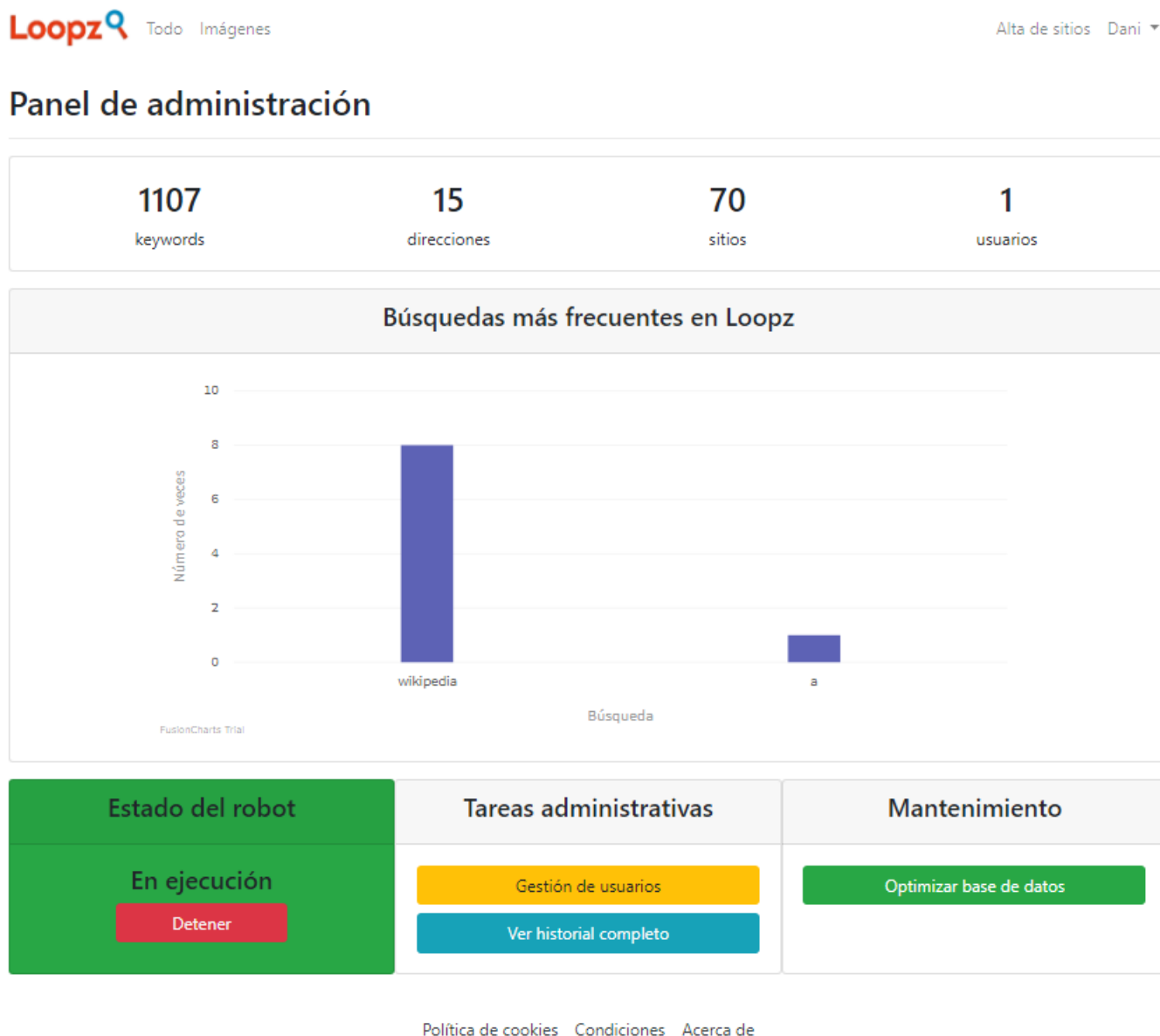
Si estás teniendo problemas haciendo click en el botón “Verificar email”, copia y pega esta dirección en tu navegador web: <https://loopz.cf/email/verify/12?expires=1557939785&signature=cb08bd88367c2e688bd44fe6daac7a8666c732cb1c4e1f6c7289df728992e7d0>

Loopz S.A. 2019

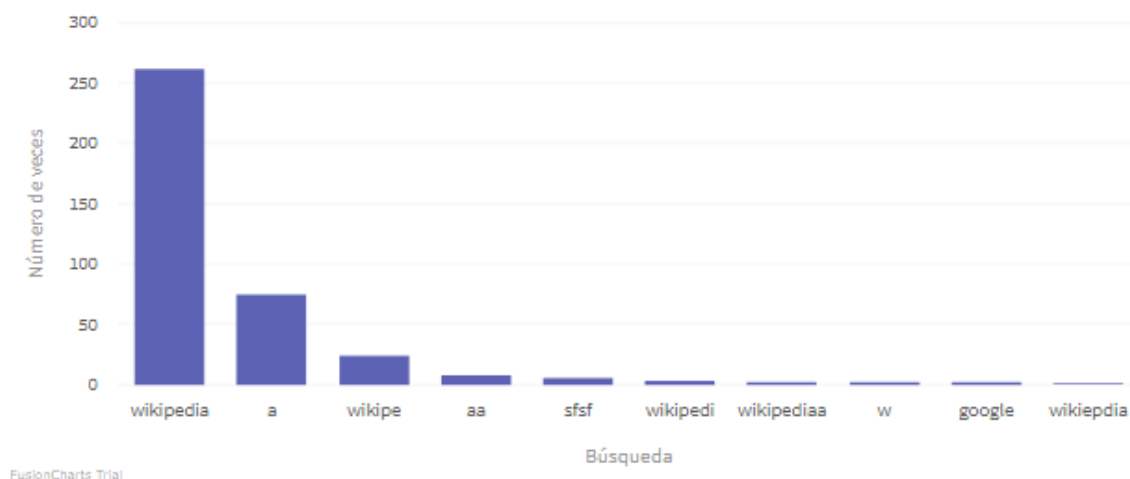


### 3.14.10. Panel de administración

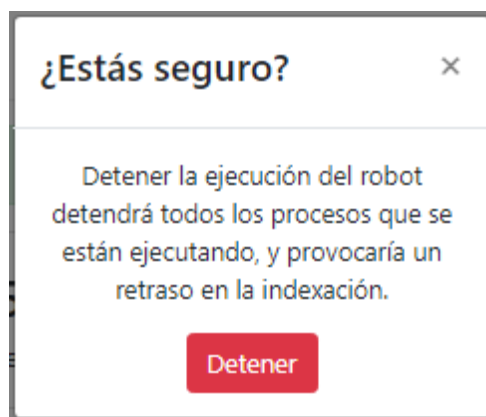
La página se ha diseñado con el fin de facilitar las tareas de administración: consultar historial de búsquedas, gestionar usuarios, consultar historial completo, y mantenimiento (optimizar la base de datos manualmente) de la página web, y la gestión del robot de indexación (comprobar su estado y ejecutarlo o detenerlo), además de ofrecer una visión global de la página web al administrador, mostrando datos y un gráfico. El diseño utiliza las cards de Bootstrap.



El gráfico se ha creado utilizando la librería FusionCharts de PHP, que genera gráficos fácilmente, con unas pocas líneas de código y un array con los datos. Este código se ha introducido en la vista Blade directamente.




Al hacer click en detener el robot, se abrirá una ventana pidiendo confirmación:



Al hacer click en Gestión de usuarios, se abre un listado de todos los usuarios registrados en Loopz divididos en 20 usuarios por página. En cada registro del listado se puede ver el nombre, email y comprobar si un usuario tiene o no verificada su cuenta (por email).

En caso negativo de que un usuario no tenga verificada su cuenta, se presenta la opción de verificarlo manualmente. Además, también aparece la opción de eliminar un usuario.


[Todo](#)
[Imágenes](#)

[Alta de sitios](#)
[Dani](#)

ID	Email	Nombre		
1	prueba@gmail.com	Dani	Verificado	<a href="#">Eliminar</a>
2	prueba3@gmail.com	d	No verificado <a href="#">[Verificar]</a>	<a href="#">Eliminar</a>
3	prueba4@gmail.com	d	Verificado	<a href="#">Eliminar</a>
4	prueba5@gmail.com	d	Verificado	<a href="#">Eliminar</a>
5	prueba6@gmail.com	d	No verificado <a href="#">[Verificar]</a>	<a href="#">Eliminar</a>
6	prueba7@gmail.com	d	No verificado <a href="#">[Verificar]</a>	<a href="#">Eliminar</a>
7	prueba2@gmail.com	d	No verificado <a href="#">[Verificar]</a>	<a href="#">Eliminar</a>

1

[Política de cookies](#)
[Condiciones](#)
[Acerca de](#)

Y, al hacer click en Ver historial completo, se abrirá un historial con el mismo diseño que el personal de cada usuario, pero contendrá todos los registros, y no tendrá botón de Limpiar historial.

### 3.14.11. Configuración y preferencias

El panel de configuración y preferencias contiene todas las opciones que necesitan los usuarios para gestionar sus cuentas. Estas son: cambiar email, nombre, contraseña, modificar sus preferencias de búsqueda de forma fija (búsqueda detallada, vistas previas), modificar tema (mostrar o no los fondos de pantalla, personalizar su visualización, subiendo nuevas imágenes y seleccionando cuales mostrar), o eliminar su cuenta.

Para su diseño también se han utilizado las cards de Bootstrap. Los tres primeros formularios se presentan en 3 columnas.

#### Configuración y preferencias

Cambiar nombre	Cambiar email	Cambiar contraseña
<p><b>Nombre actual:</b> Dani</p> <p>Nuevo nombre <input type="text"/> <input type="button" value="Confirmar"/></p>	<p><b>Email actual:</b> prueba@gmail.com</p> <p>Nuevo email <input type="text"/> <input type="button" value="Confirmar"/></p>	<p>Antigua contraseña <input type="password"/></p> <p>Nueva contraseña <input type="password"/> Repite nueva contraseña <input type="password"/></p> <p><input type="button" value="Confirmar"/></p>





#### Preferencias de búsqueda

☐ Búsqueda detallada ☐ Vistas previas


#### Modificar tema

☐ Mostrar fondos de pantalla



Selecciona una imagen...

En pantallas pequeñas, los formularios se van recolocando hasta acabar estando verticalmente, pasando por dos columnas en los dos primeros formularios, en pantallas medianas.


[Todo](#)
[Imágenes](#)

Alta de sitios
Dani

Cambiar nombre

Nombre actual: Dani

Cambiar email

Email actual: prueba@gmail.com

Cambiar contraseña

Preferencias de búsqueda

☐ Búsqueda detallada
☐ Vistas previas

Cambiar nombre

Nombre actual: Dani

Cambiar email

Email actual: prueba@gmail.com

Cambiar contraseña

### 3.14.12. Páginas informativas

Estas son la página de términos y condiciones, la de uso de cookies y la de Acerca de. Son páginas de visualización de texto, las dos primeras estructuradas verticalmente en encabezados, párrafos y listas, y la última con dos cards de Bootstrap horizontalmente.

## Términos y condiciones de Loopz

En cumplimiento de lo establecido en la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal, le informamos que, mediante la cumplimentación del presente formulario, sus datos personales quedarán incorporados y serán tratados en un fichero automatizado denominado "Base de datos Loopz", titularidad de Loopz S.A., con la finalidad de poder gestionar su solicitud, así como para mantenerle informado de futuras promociones, noticias y novedades relacionadas con el sector.

Loopz S.A. se compromete a tratar de forma confidencial los datos de carácter personal facilitados y a no comunicar o ceder dicha información a terceros. Usted podrá de forma libre y voluntaria facilitar la información que se le pide en los formularios salvo en los campos que aparecen como obligatorios. La no introducción de la información solicitada como obligatoria podrá tener como consecuencia que no pueda atenderse su solicitud.

A su vez, y en virtud de lo establecido en la Ley 34/2002, de 11 de julio, de Servicios de la Sociedad de la Información y de Comercio electrónico Loopz S.A. informa que podrá utilizar las direcciones de correo electrónico facilitadas, para remitirte información acerca de sus productos y servicios, avisos y ofertas y, en general, información de carácter comercial de interés relativa a la actividad de la empresa.

Asimismo, le informamos de la posibilidad que tiene de ejercer los derechos de acceso, rectificación, cancelación y oposición de sus datos de carácter personal de forma presencial en las oficinas de Loopz S.A., acompañando copia de DNI, o bien mediante correo postal dirigido a: Avenida de América 49, Madrid (España).

## Política de cookies

Loopz S.A. informa acerca del uso de las cookies en su página web: [www.loopz.cf](http://www.loopz.cf)

### ¿Qué son las cookies?

Las cookies son archivos que se pueden descargar en su equipo a través de las páginas web. Son herramientas que tienen un papel esencial para la prestación de numerosos servicios de la sociedad de la información. Entre otros, permiten a una página web almacenar y recuperar información sobre los hábitos de navegación de un usuario o de su equipo y, dependiendo de la información obtenida, se pueden utilizar para reconocer al usuario y mejorar el servicio ofrecido.

### Tipos de cookies

Según quien sea la entidad que gestione el dominio desde donde se envían las cookies y trate los datos que se obtengan se pueden distinguir dos tipos:

- **Cookies propias:** aquéllas que se envían al equipo terminal del usuario desde un equipo o dominio gestionado por el propio editor y desde el que se presta el servicio solicitado por el usuario.
- **Cookies de terceros:** aquéllas que se envían al equipo terminal del usuario desde un equipo o dominio que no es gestionado por el editor, sino por otra entidad que trata los datos obtenidos través de las cookies.  
En el caso de que las cookies sean instaladas desde un equipo o dominio gestionado por el propio editor pero la información que se recoja mediante éstas sea gestionada por un tercero, no pueden ser consideradas como cookies propias.

Existe también una segunda clasificación según el plazo de tiempo que permanecen almacenadas en el navegador del cliente, pudiendo tratarse de:

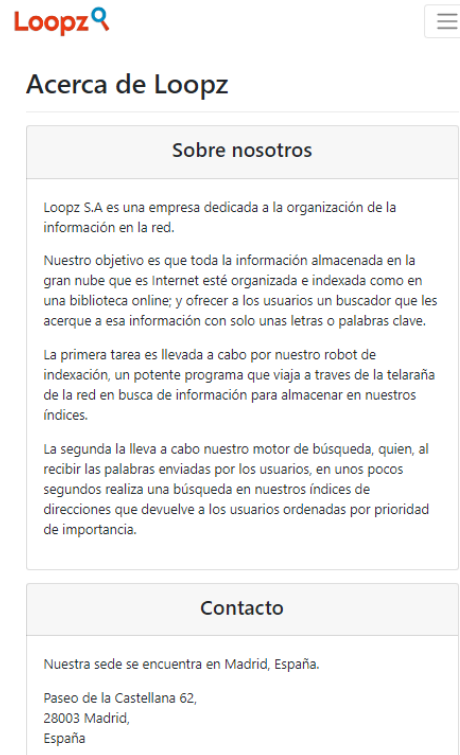
- **Cookies de sesión:** diseñadas para recabar y almacenar datos mientras el usuario accede a una página web. Se suelen emplear para almacenar información que solo interesa conservar para la prestación del servicio solicitado por el usuario en una sola ocasión (p.e. una lista de productos adquiridos).
- **Cookies persistentes:** los datos siguen almacenados en el terminal y pueden ser accedidos y tratados durante un periodo definido por el responsable de la cookie, y que puede ir de unos minutos a varios años.

Por último, existe otra clasificación con cinco tipos de cookies según la finalidad para la que se traten los datos obtenidos:

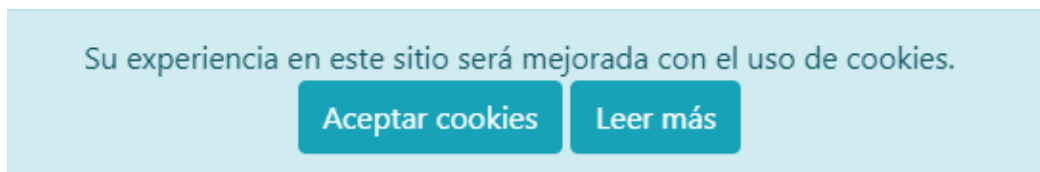
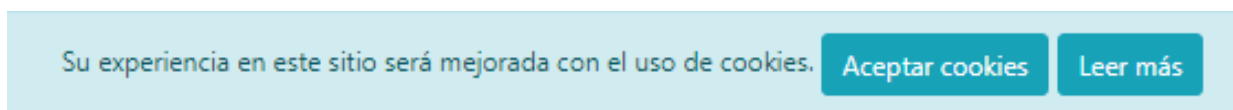
## Acerca de Loopz

Sobre nosotros	Contacto
<p>Loopz S.A es una empresa dedicada a la organización de la información en la red.</p> <p>Nuestro objetivo es que toda la información almacenada en la gran nube que es Internet esté organizada e indexada como en una biblioteca online; y ofrecer a los usuarios un buscador que les acerque a esa información con solo unas letras o palabras clave.</p> <p>La primera tarea es llevada a cabo por nuestro robot de indexación, un potente programa que viaja a traves de la telaraña de la red en busca de información para almacenar en nuestros índices.</p> <p>La segunda la lleva a cabo nuestro motor de búsqueda, quien, al recibir las palabras enviadas por los usuarios, en unos pocos segundos realiza una búsqueda en nuestros índices de direcciones que devuelve a los usuarios ordenadas por prioridad de importancia.</p>	<p>Nuestra sede se encuentra en Madrid, en la siguiente dirección:</p> <p>Paseo de la Castellana 62, 28003 Madrid, España</p> <p>Puedes contactarnos por email o por teléfono:</p> <p>Información: <a href="mailto:info@loopz.cf">info@loopz.cf</a> Director: <a href="mailto:daniel.lopez@loopz.cf">daniel.lopez@loopz.cf</a> Teléfono: 910 23 21 53 Horario: 9:00 - 18:00</p>

En pantallas pequeñas se muestran las cards verticalmente.



Para cumplir con la política de cookies, se ha instalado un plugin de Laravel llamado *laravel-cookie-consent* que implementa la ventana de aviso de cookies, fijada en la parte de debajo de la página. Esta ventana desaparece al aceptar las cookies durante la sesión actual.



En pantallas pequeñas cambia un poco su estructura, mostrando los dos botones debajo del texto.

### 3.14.13. Páginas de errores

Las páginas de errores son las que aparecen cuando las peticiones HTTP devuelven códigos de error(403, 404, 419, 500, 503, etc). Laravel ya tiene por defecto unas vistas de errores, pero se han modificado para integrarlas en la interfaz de Loopz.

En todas se ha utilizado el layout “secondary”, y se muestra el código de error en grande y en color rojo, encima del mensaje de error, centrados.

#### Error 404

La página a la que intentas acceder no existe

[Política de cookies](#) [Condiciones](#) [Acerca de](#)

#### Error 500

Error del servidor

[Política de cookies](#) [Condiciones](#) [Acerca de](#)

#### Error 503

Servicio no disponible

[Política de cookies](#) [Condiciones](#) [Acerca de](#)



## 3.14.14. Imágenes

Las imágenes utilizadas en el carrusel de la página web, así como los iconos de búsqueda, configuración, cambiar tema, eliminar, ayuda, son imágenes sin copyright obtenidas por Internet en las siguientes direcciones:

- <https://icon-icons.com>: Para los iconos
- <https://pixabay.com/es/>: Para los fondos del carrusel

El logotipo de Loopz ha sido diseñado con Adobe Photoshop CC 2018.



### 3.15. *Hosting y dominio*

Se ha utilizado el hosting gratuito Skylium, que permite la ejecución de comandos (esta función era necesaria para las búsquedas con vista previa, que ejecutan cURL), y permite el acceso remoto a las bases de datos MySQL (era necesario para poder acceder desde el programa Java).

En el hosting se aloja la página web, con el motor de búsqueda y la base de datos MySQL. El robot de indexación se aloja en un servidor Windows privado, en el cual se ejecuta la interfaz gráfica Java Swing.

Se ha utilizado esta estructura porque el plan gratuito de Skylium no permite ejecutar programas Java.

Además, se utiliza el dominio gratuito loopz.cf, obtenido en el proveedor DNS Freenom.



## 4. Modelado de la solución

Para el modelado de la solución fueron necesarios los siguientes recursos:

### 4.1. Recursos humanos

Un programador full-stack, con conocimiento de lenguajes en el backend (PHP con Laravel), de lenguajes en el frontend (HTML, CSS, JavaScript) y de Java SE, además de conocimiento sobre bases de datos relacionales SQL (MySQL).

Yo, Daniel López, he hecho la labor de este programador.

### 4.2. Recursos hardware

- ❖ Un ordenador Windows con acceso a Internet, para toda la etapa de desarrollo y pruebas.
- ❖ Un dispositivo Android con acceso a Internet, para probar la capacidad de adaptación a todos los tamaños de pantalla.

### 4.3. Recursos software

- ❖ Un entorno de desarrollo (IDE) Java: NetBeans.
- ❖ Un entorno de desarrollo (IDE) PHP: PHPStorm.
- ❖ Un editor de texto: Atom
- ❖ Lenguajes de programación/Frameworks del lado de servidor (PHP, Laravel)
- ❖ Lenguajes de programación/Frameworks del lado de cliente (JS, JQuery, Bootstrap)
- ❖ Java SE.
- ❖ Lenguajes de programación web (HTML5, CSS3).
- ❖ Un editor de imágenes: Adobe Photoshop CC 2018
- ❖ Un servidor web en local con base datos, para la etapa de desarrollo: XAMPP
- ❖ Un hosting: Skylum
- ❖ Un dominio: Freenom (.cf)
- ❖ Una base de datos relacional MySQL: Proporcionada por Skylum.
- ❖ Navegadores web (Chrome y Firefox).
- ❖ StarUML.
- ❖ Microsoft Office 2016.

## 5. *Planificación temporal*

Como la aplicación se divide principalmente en dos partes (robot de indexación, en Java, y motor de búsqueda, en PHP), se tendrá en cuenta en la planificación temporal.

Dispongo de 81 días, desde el 7 de Marzo de 2019 hasta el 27 de Mayo de 2019. Suponiendo una dedicación de 2 horas al día, se repartirá el trabajo de la siguiente forma:

- ❖ La parte del robot de indexación, programado en Java, llevará un tiempo de desarrollo de 1 mes:  
 $30\text{días} \times 2\text{h} = 60\text{ horas}$
- ❖ La parte del motor de búsqueda y la página web, programado en PHP (Laravel), llevara también un tiempo de 1 mes de desarrollo:  
 $30\text{días} \times 2\text{h} = 60\text{ horas}$
- ❖ El tiempo restante ( $81\text{ días} - 60\text{ días} = 21\text{ días}$ ), intercalado a lo largo de las otras dos partes del desarrollo, será dedicado a realizar la memoria de proyecto, y finalmente la exposición en Microsoft Office PowerPoint:  
 $21\text{días} \times 2\text{h} = 42\text{ horas}$

Horas en total=  $60 + 60 + 42 = 162\text{ horas}$ .

## 6. *Costes del proyecto*

El proyecto ha llevado un tiempo de desarrollo de 162 horas, en un periodo de 3 meses, con jornadas de 2 horas al día cada día de la semana.

Al ser una aplicación de gran tamaño, con programación en varios lenguajes (Java SE, PHP, JS, HTML5 y CSS3), además de la utilización de varios frameworks (Laravel, JQuery, Bootstrap), y la labor de investigación que se realizó para llegar a una solución de robot de indexación y motor de búsqueda que fueran eficaces y eficientes (en cuanto a tiempo), el precio por hora del desarrollo asciende a 40€.

Por lo tanto:

$$2 \text{ horas} \times 81 \text{ días} \times 40\text{€} = 6.480\text{€}$$

El precio total de la aplicación, con un desarrollo de 81 días, es 6.480€

En cuanto al mantenimiento, es requerida una figura de administrador, que gestione los usuarios, el funcionamiento del robot de indexación y que realice la optimización de la base de datos en caso de ser necesaria; además se solventar problemas que puedan surgir.

El precio de este mantenimiento sería 1200€ al mes.

## 7. Conclusiones finales

### 7.1. Grado de cumplimiento de los objetivos fijados.

- ❖ He creado una aplicación cuyo motor de búsqueda es capaz de buscar entre las páginas web registradas en su base de datos por el robot de indexación.
- ❖ He incluido en la página web un formulario a través del cual los usuarios pueden dar de alta sitios web en la base de datos sin tener que esperar a que el robot los encuentre en la red.
- ❖ El funcionamiento, así como la coordinación Motor de búsqueda-Robot de indexación es totalmente transparente para el usuario, de forma que el usuario simplemente tiene que entrar en la web y hacer búsquedas, sin ser consciente de la complejidad que hay detrás.
- ❖ Se ha implementado el registro de usuarios con el fin de permitir el acceso a más funciones, como el historial de búsqueda personal, las preferencias de búsqueda, personalización de la página principal, o la posibilidad de dar de alta nuevos sitios web.
- ❖ Los usuarios registrados tienen acceso a un historial de búsqueda personal, con la posibilidad de eliminarlo.
- ❖ También se ha implementado la indexación y búsqueda de imágenes en la red.
- ❖ Se ha establecido conexión entre la parte Java y la parte PHP de la aplicación.
- ❖ Se ha implementado una interfaz gráfica sencilla en el programa Java.

En conclusión, diría que los objetivos fijados se han cumplido al 100% y se han añadido más funcionalidades de las planificadas en el inicio del desarrollo.

## 7.2. Propuesta de modificaciones o ampliaciones futuras del sistema implementado.

Con el fin de mejorar la aplicación, y de añadir nuevas funciones útiles para los usuarios, se plantean una serie de modificaciones y ampliaciones de la aplicación el futuro:

- ❖ Mejora del programa Java:
  - Mejora de la interfaz Java Swing, para mostrar más información de los Threads que se están ejecutando.
  - Implementar la gestión de la aplicación a través de comandos, con el fin de poder ejecutarla en sistemas Linux sin interfaz gráfica.
- ❖ Mejora de los algoritmos de indexación y búsqueda de imágenes:
  - Con el fin de hacer esta más precisa, tener en cuenta el contenido de las propias imágenes (formas, colores, tamaños, etc) y no solo las palabras que tengan alrededor.
- ❖ Mejora del algoritmo de indexación de palabras:
  - Establecer una prioridad en función del contenido general de la página y no solamente por el número de veces que aparece en la página web
  - Implementar una función que pueda diferenciar las palabras indexadas por su existencia o no en el diccionario (de cualquier idioma), influyendo esto en la prioridad que se les atribuye.
  - Tener en cuenta la popularidad de la página (cantidad de visitas) a la hora de establecer las prioridades.
  - Añadir un nuevo campo idioma de las palabras.
- ❖ Mejora del motor de búsqueda:
  - Dar más prioridad de aparición en los resultados a las páginas más visitadas por los usuarios.
  - Mostrar en el desplegable de autocompletar las palabras ordenadas por mayor popularidad (palabras más buscadas).
  - Optimizar aún más el algoritmo, para igualar los tiempos de búsqueda con vistas previas a los que se tarda ahora sin ellas.
  - Permitir hacer búsquedas por idiomas.
- ❖ Mejora de la página web:
  - Traducir la web a más idiomas que el español.
- ❖ Mejora del despliegue:
  - Utilización de un servidor VPS sin límite de recursos, con el fin de alojar la página web, base de datos y robot de indexación en el mismo servidor.
  - Cambio a un dominio de pago .com o .es.



## 8. Bibliografía y Webgrafía

- <https://parzibyte.me/blog/2017/11/07/proteger-controlar-restringir-acceso-a-imagenes-usando-php-apache/>
- <https://styde.net/como-personalizar-el-correo-de-recuperacion-de-contrasenas-en-laravel-5-3-y-5-4/>
- [https://www.youtube.com/channel/UCdulls-x\\_xrRd1ezwJZR9ww](https://www.youtube.com/channel/UCdulls-x_xrRd1ezwJZR9ww)
- <https://www.haztupaginaweb.com/certificado-ssl-gratis/>
- <https://laracasts.com>
- <https://stackoverflow.com/>
- <https://www.nigmacode.com/laravel/Subir-proyecto-laravel-a-hosting>
- <https://laravel.com/docs/5.8/>
- <https://getbootstrap.com/docs/4.0/getting-started/introduction/>
- <https://api.jquery.com/>
- <https://github.com/crawler-commons/crawler-commons>
- <https://developers.virustotal.com/reference>
- <https://ayudaleyprotecciondatos.es/modelo-politica-cookies/>
- <https://icon-icons.com>
- <https://pixabay.com/es/>
- <https://www.vivablogger.com/convertir-utf8-a-utf8mb4/>
- <https://steemit.com/technology/@ndnthor/como-convertir-una-base-de-datos-antelope-a-barracuda-desde-phpmyadmin>
- <https://www.php.net/manual/es/>
- <https://parzibyte.me/blog/2017/04/17/cambiar-zona-horaria-timezone-laravel-5-4-php/>
- <https://www.w3schools.com/bootstrap4/default.asp>
- <https://desarrolloweb.com/manuales/manual-laravel-5.html>
- <https://www.calculadorafreelance.com/>
- <https://fluidattacks.com/web/es/defends/java/detener-hilos/>
- <https://www.jose-aguilar.com/blog/autocompletar-campo-con-jquery-ajax-y-php/>
- <https://www.lawebdelprogramador.com/>
- <https://jsoup.org/>
- <https://mvnrepository.com/>
- <https://commons.apache.org/>
- <https://github.com/google/gson>
- <https://codepen.io/>
- <https://www.lucidchart.com/>
- <https://docs.oracle.com/javase/7/docs/api/>

- <https://www.fusioncharts.com/php-charts>
- <https://www.galisteocantero.com/buscar-el-valor-mas-repetido-en-sql/>
- <https://github.com/spatie/laravel-cookie-consent>
- <http://panamahitek.com/crear-un-archivo-jar-con-librerias-externas/>