



## Trabajo Fin de Grado

Software de seguimiento de posiciones en  
Google “Serptracker”

Autor

D. Jorge Aranda Barbero

Directora

Dra. Raquel Lacuesta

Escuela Universitaria Politécnica Teruel  
Universidad de Zaragoza  
2019



## Tabla de contenido

<b>1. Introducción y objetivos .....</b>	<b>4</b>
<b>2. Estado del arte .....</b>	<b>5</b>
<b>2.1. Lenguajes de scripting .....</b>	<b>5</b>
<b>2.1.1. Ruby .....</b>	<b>6</b>
<b>2.1.2. Perl .....</b>	<b>7</b>
<b>2.1.3. Python .....</b>	<b>8</b>
<b>2.2. Lenguajes de programación.....</b>	<b>9</b>
<b>2.2.1. NodeJS.....</b>	<b>9</b>
<b>2.2.2. AngularJS.....</b>	<b>10</b>
<b>2.2.3. Laravel.....</b>	<b>11</b>
<b>2.3. Lenguajes de marcado.....</b>	<b>12</b>
<b>2.4. Estudio de otras aplicaciones similares a Serptracker .....</b>	<b>13</b>
<b>2.4.1. Tabla comparativa .....</b>	<b>16</b>
<b>2.5. ¿Por qué Ruby y NodeJS? .....</b>	<b>17</b>
<b>2.6. ¿Qué es el web scraping? .....</b>	<b>18</b>
<b>2.6.1. Librería Nokogiri .....</b>	<b>18</b>
<b>3. Fase de análisis y diseño.....</b>	<b>19</b>
<b>3.1. Descripción general .....</b>	<b>19</b>
<b>3.1.1. Perspectiva y objetivo del proyecto .....</b>	<b>19</b>
<b>3.2. Requisitos funcionales .....</b>	<b>20</b>
<b>3.3. Requisitos no funcionales.....</b>	<b>22</b>
<b>3.4. Casos de uso .....</b>	<b>23</b>
<b>3.5. Diagrama entidad relación .....</b>	<b>24</b>
<b>3.6. Diagrama de actividad.....</b>	<b>25</b>
<b>3.6.1. Obtención de proxys .....</b>	<b>25</b>
<b>3.6.2. Obtención de posiciones .....</b>	<b>26</b>
<b>3.7. Diseño de prototipos de baja fidelidad .....</b>	<b>27</b>
<b>4. Fase de desarrollo .....</b>	<b>31</b>
<b>4.1. Análisis del funcionamiento del motor de búsqueda Google .....</b>	<b>31</b>
<b>4.2. Búsqueda de solución para realizar una consulta de manera automática .....</b>	<b>32</b>
<b>4.3. Obtención de datos de proxys de manera masiva.....</b>	<b>33</b>
<b>4.4. Verificación de proxys válidos .....</b>	<b>34</b>
<b>4.5. Almacenamiento de resultados en base de datos MySQL.....</b>	<b>34</b>
<b>4.6. Desarrollo de aplicación con NodeJS .....</b>	<b>34</b>
<b>4.7. Elección del servidor donde alojar Serptracker .....</b>	<b>36</b>
<b>4.8. Configuración del servidor y ejecución de Serptracker.....</b>	<b>37</b>
<b>4.9. Configuración del dominio .....</b>	<b>37</b>
<b>5. Fase de pruebas: caso práctico .....</b>	<b>38</b>

5.1.	Introducción.....	38
5.2.	Creación del proyecto .....	38
5.2.1.	Elección del nicho .....	38
5.2.2.	Keyword research .....	38
5.2.3.	Instalación y configuración de la web .....	39
5.2.4.	Indexación en Google .....	39
5.3.	Seguimiento de resultados con Serptracker.....	39
5.4.	Conclusiones de la prueba en Serptracker .....	41
6.	<i>Accesibilidad y Usabilidad</i> .....	42
6.1.	Accesibilidad .....	42
6.2.	Usabilidad .....	42
7.	<i>Licencia Documental</i> .....	43
8.	<i>Ley de protección de datos</i> .....	44
9.	<i>Conclusión</i> .....	45
10.	<i>Referencias bibliográficas</i> .....	46
11.	<i>Anexos</i> .....	48
11.1.	MANUAL DE USUARIO.....	48
11.1.1.	Registro .....	48
11.1.2.	Login .....	49
11.1.3.	Añadir nuevo proyecto .....	50
11.1.4.	Editar proyecto existente .....	51
11.1.5.	Eliminar un proyecto.....	51
11.1.6.	Tipos de gráficas.....	52

## Resumen

El presente documento expone todo el proceso de realización del proyecto “Software de seguimiento de posiciones en Google “Serptracker””, desde la fase de análisis de necesidades actuales en el ámbito de herramientas SEO hasta la fase de implementación y pruebas de la herramienta desarrollada.

La finalidad de este proyecto es el desarrollo de una herramienta que permita solventar la problemática actual de las agencias de marketing digital o *freelances* para mejorar el seguimiento del posicionamiento de los sitios web.

La herramienta permite realizar un seguimiento diario de las posiciones de las palabras clave a posicionar en Google, además de realizar un análisis inicial del SEO *Onpage* del proyecto. Gracias a esto, se puede valorar si las acciones llevadas a cabo son de utilidad o, si, por el contrario, desfavorecen el posicionamiento.

Mediante este seguimiento podría detectarse rápidamente la eficiencia de la estrategia seguida y decidir si continuar con ella o no.

Esta herramienta de análisis también surge por la necesidad de encontrar una alternativa gratuita que realice estos análisis, ya que, en la actualidad, las existentes tienen un precio elevado.

Como resultado, esta herramienta podría suponer un ahorro económico para agencias de posicionamiento y la mejora en el seguimiento y posicionamiento de cualquier sitio web. Ya que, si se observa que la evolución de un proyecto está perdiendo posiciones en Google, podrán detectar rápidamente el problema y realizar las modificaciones oportunas para mejorarla.

## Palabras clave

Serptracker, SEO, posicionamiento web, Google.

## Abreviaturas

Siglas	Significado
SEO	Search Engine Optimization
H1	head 1
H2	head 2
CRUD	Create, Read, Update, Delete
RPM	Revenue Per Thousand impressions
API	Application Programming Interface
CMS	Content Management System
E-R	Entidad Relación

## Glorario

**Search Engine Optimization:** optimización de una web para mejorar su visibilidad en los motores de búsqueda.

**Scraper:** herramienta dedicada a recoger masivamente datos.

**Bulk:** volcado masivo de datos.

**SEO Onpage:** serie de pautas básicas internas que hacen que una web esté bien estructurada o no a ojos de Google.

**Responsive:** adaptabilidad de los elementos de diferentes resoluciones de manera automática.

**Framework:** estructura tecnológica con módulos que facilitan el desarrollo software.

**SEO:** optimización del posicionamiento web en los motores de búsqueda para mejorar la visibilidad de una página web en los buscadores.

**Bootstrap:** framework para crear interfaces web mediante HTML y CSS que sean responsive.

**MySQL:** sistema de gestión de base de datos relacional.

**Keyword:** son las palabras clave por las que se quiere conseguir que la web tenga visibilidad en los motores de búsqueda.

**RPM:** remuneración por cada mil impresiones de anuncios en un sitio web.

**API:** se trata de una interfaz de programación de aplicaciones que contiene una serie de funciones y subrutinas orientadas a facilitar y reducir la programación de una aplicación.

**Trial:** periodo de prueba.

**Adwords:** plataforma de Google para comprar publicidad en su buscador.

**CMS:** referido a gestores de contenido principalmente para páginas web.

## Agradecimientos

En primer lugar, agradecer a mi tutora Raquel Lacuesta la oportunidad de realizar el trabajo final de carrera sobre un tema que me apasiona y al que he dedicado tiempo desde antes incluso de empezar la carrera.

Agradecer también a mis amigos del sector por sus valoraciones y opiniones de mejora de la herramienta. Y también a todos mis compañeros de la diputación que me han apoyado durante el desarrollo del trabajo de fin de grado.

Por último, agradecer a mi familia el apoyo y ánimos que me han dado durante estos años en la Universidad Politécnica de Teruel.

## 1. Introducción y objetivos

El SEO se define como el conjunto de acciones de una página web orientadas a mejorar su visibilidad en los motores de búsqueda (como Google), es decir, intentar posicionar una web en las primeras posiciones del buscador.

La aplicación desarrollada se llama Serptracker y permite realizar un seguimiento diario de la evolución de las posiciones de una web en Google. Para la obtención y procesado de los datos se utiliza el lenguaje de programación Ruby, y para visualizar los datos se utiliza una web desarrollada en NodeJS.

Las agencias actuales de marketing digital enfocadas en llevar el SEO de las páginas de sus clientes tienen la necesidad de mostrar gráficamente [1] los avances de posiciones que van obteniendo a modo de resultado. Estos informes sirven para que el cliente pueda ver rápidamente los progresos que están haciendo y valorar si es resultado del trabajo es suficiente o no.

No obstante, también existen *freelancers* o personas con una o varias webs propias que quieren ver estos resultados, pero que no están dispuestos a pagar el coste de las herramientas profesionales que las agencias más grandes sí podrían permitirse. Estos precios tan elevados se deben, en gran medida, al coste de la API que ofrece Google para mostrar los resultados de búsqueda.

Una empresa busca posicionarse por una o varias palabras clave, hacer esa búsqueda a diario y almacenar los resultados supone un coste de esa API. Serptracker, ofrece eso mismo, pero de manera gratuita.

La manera en la que la herramienta Serptracker consigue los resultados es mediante búsquedas, pero con la gran diferencia de que utiliza diferentes *user-agents* [11] y *proxys* en cada una de las búsquedas, por lo que Google no detecta que es un proceso automático y no hace saltar su *captcha* para validar que “no eres un robot”. Esto es necesario debido a que se deben producir numerosas búsquedas iguales de manera diaria.

La aplicación está desarrollada con numerosas tecnologías, destacando NodeJS para mostrar y analizar datos y Ruby como lenguaje de scripting para obtención de resultados. La base de datos se basa en MySQL.

Los objetivos a grandes rasgos, después de esta pequeña introducción, serían:

- 1- Conseguir que pequeñas empresas, agencias de marketing o usuarios comunes lleguen a poder ver la evolución de posiciones en Google de sus proyectos (webs) de una manera gratuita. Una vez estén realizando el seguimiento, podrán ver si la estrategia seguida les favorece o no en el alcance de la posición número uno para las palabras clave elegidas.
- 2- Permitir observar de manera rápida el *SEO Onpage* de la página principal del proyecto añadido en la herramienta.

En este documento se describen todos los pasos seguidos para el desarrollo de este proyecto, desde la fase de análisis inicial hasta el diseño e implementación de este.

En primer lugar, se explican de manera global las tecnologías utilizadas para la realización de este TFG, así como información relevante sobre las mismas y el motivo de su elección. Posteriormente, se explica el proceso de diseño de la estructura y funcionamiento, y, finalmente, cómo y dónde se ha decidido implementarlo.

## 2. Estado del arte

Actualmente, el SEO es uno de los puntos más importantes para tener en cuenta cuando se posee un negocio. La diferencia entre el éxito y el fracaso puede estar simplemente en la visibilidad que tenga el mismo en Google. Estar una posición más arriba o más abajo puede marcar la diferencia.

Como ejemplo, y según numerosos estudios, el 60% de los clicks de una búsqueda se los lleva el primer resultado. Y más del 90% de los clicks se reparten solo en la primera página de Google (10 primeros resultados). Es por ello por lo que debemos buscar siempre la mejor posición en este motor de búsqueda que domina el mercado, por eso es tan importante el SEO, porque un buen SEO se traduce en mejor alcance de posibles clientes.

Las diferentes estrategias que se utilizan para mejorar el SEO de una web van desde el SEO *Onpage* hasta el SEO *Offpage* y *link building*. Serptracker permite ver y analizar rápidamente si las estrategias que se están utilizando están funcionando o, si, por el contrario, desfavorecen las posiciones en Google.

Las tecnologías que se usan para lograr este cometido son **NodeJS** y **Ruby**. A continuación, se pasará a valorar las diferentes alternativas y el por qué de esta elección.

### 2.1. Lenguajes de scripting

Un lenguaje de *scripting* es un lenguaje de programación en el que el código se interpreta durante el tiempo de ejecución, es decir, que no es compilado por el procesador del ordenador. Un *script* requiere un intérprete y un programa requiere un compilador.

Se engloba a cualquier lenguaje que esté destinado a comunicarse e integrarse con otros lenguajes de programación. Suelen utilizarse en conjunto a otros lenguajes de programación para así completar su cometido. Un ejemplo sería la combinación de PHP con HTML, es decir, la combinación de un lenguaje de *scripting* y uno de marcado.

### 2.1.1. Ruby



Ruby es un lenguaje de *scripting* que permite realizar tareas de manera muy rápida. Existen dos tipos que podrían llevar a confusión: ruby y ruby *on the Rails*.

Ruby *on the Rails* es un *framework* muy popular que se utiliza para desarrollos web. En este caso, para esa tarea, se ha elegido NodeJS.

Se ha escogido Ruby como lenguaje de *scripting*. Ruby es un lenguaje orientado a objetos creado por Yukihiro Matsumoto en 1993. Se trata de un lenguaje interpretado.

Posee multitud de librerías llamadas “gemas”. Se instalan de manera muy sencilla y suelen tener una gran documentación que ayuda a comenzar a utilizarlas. Algunas de estas librerías se han utilizado para facilitar el *web scraping* de *proxys* y resultados de búsqueda (recolección de manera masiva de *proxys* y resultados de búsquedas en Google).

Algunas ventajas de Ruby son:

- Sintaxis simple y consistente.
- Permite realizar llamadas directamente al Sistema Operativo.
- Lenguaje basado en objetos.
- No es necesario declarar variables.
- Es un lenguaje de programación de alto nivel, multiplataforma y *opensource*.
- Posee una gran comunidad que desarrolla continuamente nuevas librerías con la que facilitar las tareas.

Desventajas del lenguaje Ruby:

- En ocasiones es lento al procesar el lenguaje.
- Continúa en desarrollo.

### 2.1.2. Perl



Perl es un lenguaje de scripting. Significa “*Practical Extracting and Reporting Language*”. Comúnmente utilizado para extraer información de ficheros y realizar informes con ellos.

Fue creado por Larry Wall en el año 1987, con el propósito de hacer de manera automática tareas administrativas que eran repetitivas.

Como ventajas, hay que destacar que posee licencia GPL, lo cual hace que sea un lenguaje gratuito de utilizar, además de ser multiplataforma.

Con la llegada de nuevos lenguajes, Perl ha decaído en popularidad, aunque se sigue utilizando para realizar scripts en servidores de manera que automatizan o faciliten ciertas tareas.

Otra ventaja muy destacable que posee es que puede conectarse con multitud motores de base de datos sin ningún problema y de una manera muy sencilla. Tiene la posibilidad de ejecutar subrutinas en otros lenguajes de programación.

Es un lenguaje de programación con una curva de aprendizaje fácil, ya que se trata de un lenguaje de alto nivel. Es eficiente y permite utilizarse para muchos propósitos, como por ejemplo entornos web o servidores.

Una posible desventaja sería que solo posee tres tipos diferentes de datos:

- **El tipo de datos lista:** se trata de una colección de tipos de datos escalares (*array*).
- **El tipo de datos escalar:** se trata de un entero o una cadena.
- **El tipo de datos hash:** común a otros lenguajes, consiste en asociar valores con claves.

Otra desventaja sería que se trata de un lenguaje interpretado. Significa que se compila al principio de su ejecución, lo cual puede repercutir en que el curso de ejecución sea más lento si lo comparamos con otros lenguajes de programación.

Además de esto, carece de un buen sistema que permita capturar y tratar bien las excepciones, por lo que la ejecución puede interrumpirse en cualquier momento y no continuar.

### 2.1.3. Python



Python es un lenguaje de scripting muy popular debido a la sencillez de su sintaxis. Comparte numerosas características con Perl.

El gran porcentaje de usuarios que utilizan python podrán dividirse entre gente que utiliza python para programar páginas HTML con Django, y gente que lo utiliza para hacer scripts que automatizan tareas.

Python fue creado a finales de 1980 por Guido van Rossum, el cual publicó la primera versión hasta el año 1991. Durante esos años, la programación en Unix se realizaba en gran medida en C, Python surgió por la necesidad de crear algo más sencillo con lo que programar.

Este lenguaje es de propósito general, es decir, no se encuentra orientado a un fin concreto. Pueden llegar a crearse desde páginas web, hasta software multiplataforma.

Orientado a objetos, con las ventajas que ello supone, como por ejemplo la reutilización de código, la herencia o el polimorfismo.

Además, es multiparadigma, porque, aunque la orientación a objetos sea su punto fuerte, también posee otros estilos de programación como la programación funcional o la imperativa.

Se trata de un lenguaje muy rápido y eficiente, esto se debe a que es un lenguaje interpretado en el que el código fuente no se compila a código máquina, sino que el intérprete ejecuta el programa directamente siguiendo el código. Esto, a diferencia de lo que se pueda pensar, hace que python sea más rápido que muchos otros lenguajes de la competencia.

También se trata de tipado dinámico, por lo que cuando se declara una variable no hace falta añadirle el tipo de dato, ahorrando así tiempo al programador. Esto hace de Python un lenguaje con una curva de aprendizaje fácil, pero sacrificando un poco la optimización de la memoria.

Otra principal ventaja de este lenguaje de programación es que es de libre distribución, existe una gran comunidad de usuarios y librerías y tiene soporte para multitud de bases de datos.

Algunos programas destacables programados con Python son: Calibre y BitTorrent.

## 2.2. Lenguajes de programación

Los lenguajes de programación son un conjunto de estructuras y algoritmos destinados a decirle a un ordenador qué y cómo realizar una cierta tarea. Se compone de reglas sintácticas, semánticas y símbolos, las cuales son utilizadas para construir programas mediante instrucciones y relaciones lógicas [6].

### 2.2.1. NodeJS



NodeJS proporciona un entorno abierto multiplataforma, que trabaja en el lado del servidor y está basado en Javascript. Su ejecución es muy rápida y se trata de código abierto. Hace unos años se utilizaba Javascript en solo en la parte del cliente, no obstante, NodeJS busca que se trabaje tanto en el cliente como en el servidor.

Las principales ventajas de NodeJS son las siguientes:

- Lenguaje similar a Javascript, debido a que se basa en él. Esto facilita a los programadores a adaptarse rápidamente a él.
- Gran rendimiento y gestión de recursos. Se trata de un lenguaje de programación muy potente que maneja muy bien la gestión de hilos [9].
- Es multiplataforma, y se puede implementar en la gran mayoría de servidores.
- Permite crear aplicaciones escalables que puedan llegar a manejar miles de conexiones simultáneas sin ningún problema. Soporta muy bien los altos volúmenes de tráfico.
- Tiene multitud de librerías que facilitan la programación. Por ejemplo, la librería MySQL para el manejo de base de datos o express [4] para programación mediante plantillas.
- Es muy útil para aplicaciones en tiempo real.

Algunas desventajas de NodeJS:

- Tipos dinámicos: JavaScript permite enviar un objeto sin definir, por lo que se hace complicado a veces manejar los datos.
- Los *callbacks* hacen en muchas ocasiones que el código sea más difícil de seguir.

Algunas de las empresas online más importantes están desarrolladas con NodeJS, tales como: Ebay, Paypal o Linkdin

Alternativas a considerar

Algunas alternativas para el desarrollo de esta aplicación fueron Perl, Python, AngularJS o Laravel.

## 2.2.2. AngularJS



AngularJS es un *framework* de código abierto creado por Google con patrón MVC (Modelo Vista Controlador) lanzado por primera vez en el año 2009 y orientado a realizar aplicaciones SPA (*Single Page Applications*). Está basado en Javascript y centrado en desarrollos web *Front End*.

Este framework nació de la idea de unificar en un solo patrón MVC todo el código, ayudando así a crear código más fácil de seguir y reutilizable.

Otra de las ventajas que prometía fue la de sincronizar automáticamente los modelos y las vistas y reducir el código JavaScript.

Entre otras ventajas, también destacan sus plantillas, directivas, enlace de datos bidireccional, inyecciones de dependencia y sus pruebas.

Las desventajas de angular residen en los fallos de rendimiento y en que todos los usuarios que necesiten consumir una aplicación con Angular tienen que tener soporte de Javascript habilitado. Los fallos de rendimiento se dan porque las aplicaciones creadas con este framework son dinámicas.

Un ejemplo de aplicación desarrollada con este *framework* es Ionic.

### 2.2.3. Laravel



Laravel es un Framework PHP de código abierto basado en el patrón de diseño MVC. Su filosofía de programación reside en desarrollar código simple y reutilizable.

Tiene mucha documentación y una gran comunidad activa detrás de él. Otra ventaja, y por lo que multitud de programadores escogen este framework, es debido a que posee un sistema de plantillas muy visual y sencillo llamado Blade. Esta característica permite extender las plantillas creadas y utilizar código PHP entre todas ellas, de manera que agiliza mucho la programación de aplicaciones.

Otras características destacables que incluye son: Eloquent (ORM que incluye Laravel para manejo de base de datos, además ofrece protección contra inyecciones SQL), Routing (sistema de gestión de rutas) y *middlewares* (controladores que se ejecutan antes y después de las peticiones al servidor, de manera que permiten validar e incluir lógica previa antes de la ejecución de un proceso).

Laravel posee también muchas herramientas para probar las aplicaciones, tales como los Tests Unitarios. Estas pruebas ahorran el tiempo de “prueba y error” previo al lanzamiento de una aplicación.

En cuanto a las desventajas, destacan la dificultad de aprendizaje de las herramientas básicas para realizar una aplicación con este framework, tales como Composer o Artisan. Todas las librerías se instalan desde Composer.

El gran aliciente para escoger Laravel sería entonces la escalabilidad, facilidad de mantenimiento y la integración que ofrece respecto a un desarrollo nativo de aplicaciones web.

### 2.3. Lenguajes de marcado

Los lenguajes de marcado se refieren a la apariencia. Son utilizados para representar datos de manera gráfica. Un claro ejemplo sería el lenguaje HTML o el XML. Se utilizan mediante etiquetas y son legibles porque se encuentran en texto plano y no compilados.

Existen multitud de ellos, pero cada uno destinado a una función específica. Como ejemplo, HTML suele usarse junto con CSS para dar estructura y apariencia a un sitio web, mientras que XML es un lenguaje de mercado más enfocado en almacenar datos estructurados.

En este proyecto se utilizará el lenguaje de marcado HTML junto con CSS (*Cascading Style Sheets*), no obstante, mediante plantillas, ya que se utilizarán diversos *plugins* de NodeJS para agilizar la programación.

## 2.4. Estudio de otras aplicaciones similares a Serptracker

En un primer momento, se realizó un estudio de las posibles alternativas que te permitiesen realizar un seguimiento de palabras clave en Google. Ninguna de las que operaban eran gratuitas, solo ofrecían una prueba gratuita de 14 días y con un límite de búsquedas diarias.

Algunas de las alternativas populares encontradas son las siguientes:

- **Serprobót:** <https://www.serprobót.com/>

Esta herramienta tiene un precio de 10\$ mensuales y está limitada a 300 búsquedas diarias. Es decir, con las opciones disponibles solo podrían rastrearse 300 palabras divididas en varios proyectos, algo muy limitado si se trata de una agencia con más de un cliente.

Esta web presume de ser la más barata en cuanto a seguimiento de palabras clave en Google.

Ofrece una búsqueda gratuita de una web y 5 palabras clave, pero sin realizar seguimiento y solventando un *reCaptcha* de Google.

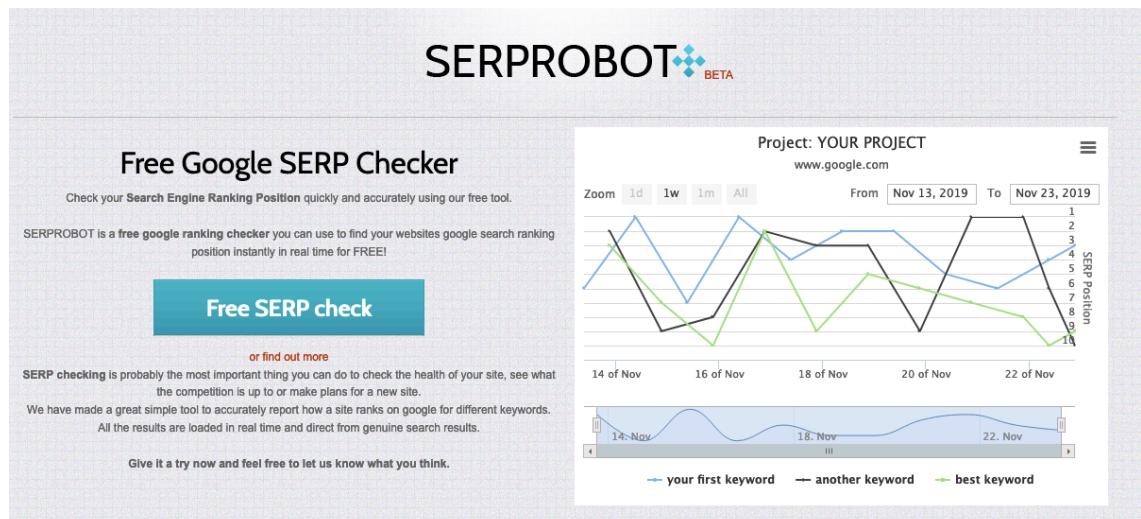


Figura 2.4.1. Página web Serprobót

- **Seranking:** <https://seranking.com/es/>

Ofrece una prueba gratuita muy limitada y durante 7 días. Los precios parten a partir de 29€ al mes con un rastreo de máximo 250 palabras clave repartidas en 10 proyectos diferentes.



Figura 2.4.2. Página web Seranking

- **Accuranker:** <https://www.accuranker.com/>

Esta herramienta ofrece una prueba trial de 14 días. Ofrece un seguimiento de 500 palabras clave al mes por un precio de 49\$. No permite segmentar la localización de la búsqueda, por lo que si eres de España los resultados no serían del todo fiables al tratarse de un seguimiento en Google con idioma inglés.

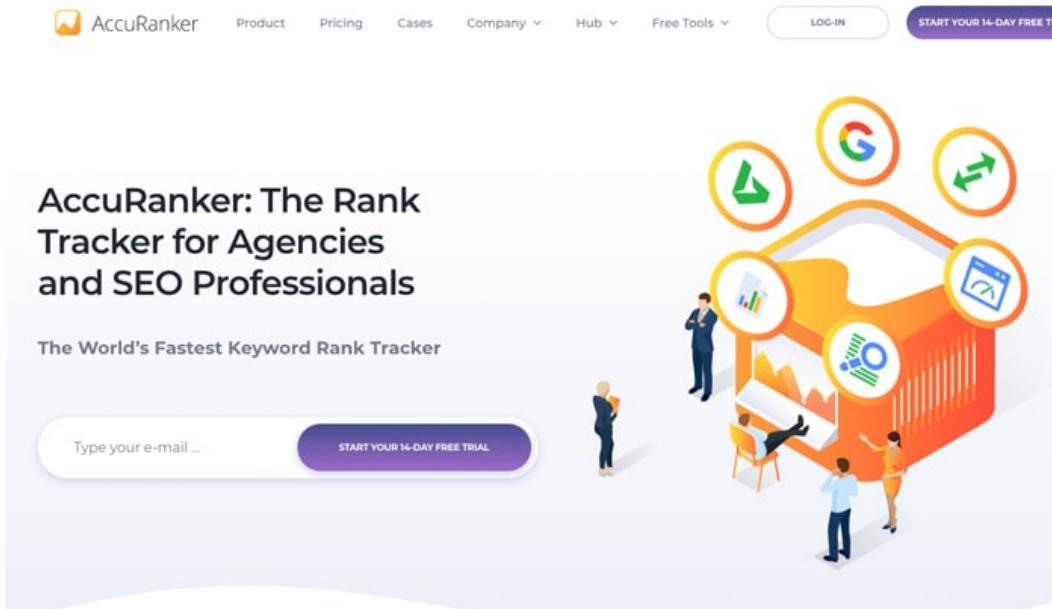


Figura 2.4.3. Página web Accuranker

- **Serpstate:** <https://serpstat.com/>

Permite elegir el buscador del país de donde se quiere hacer el seguimiento de posiciones, pero los precios varían, siendo el pack más básico de 500 palabras clave a rastrear por un precio de 69\$ al mes.



Figura 2.4.4. Página web Serpstate

- **SerpWatcher:** <https://serpwatcher.com/>

Poseen una prueba trial de 10 días. Los precios son a partir de 49\$ al mes por monitorear 200 palabras clave.

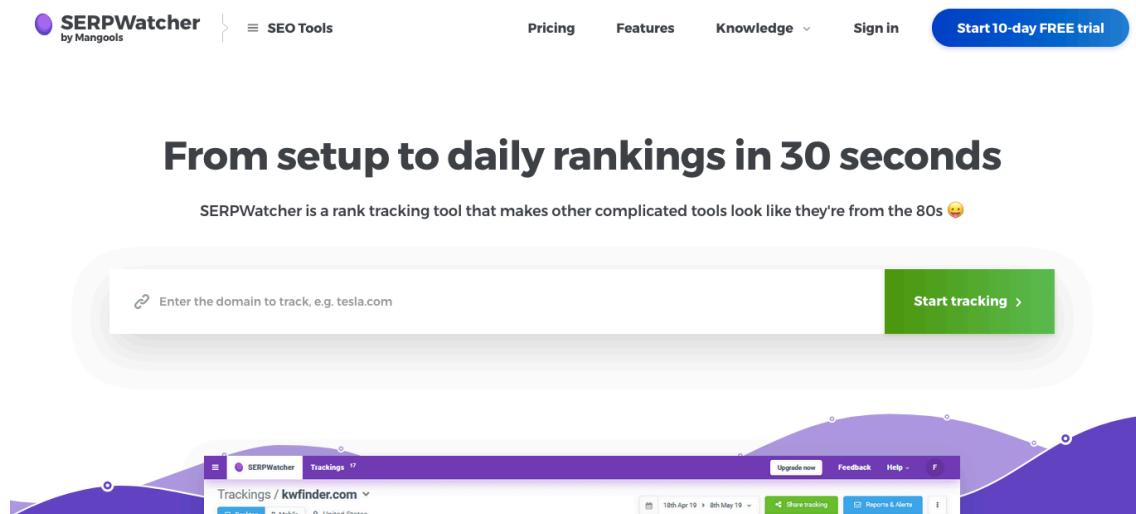


Figura 2.4.5. Página web Serpwatcher

- **Proranktracker:** <https://proranktracker.com/>

Poseen un periodo de prueba de 7 días, con un seguimiento máximo de 20 palabras clave. Los precios van a partir de 25\$ al mes por 200 palabras clave.

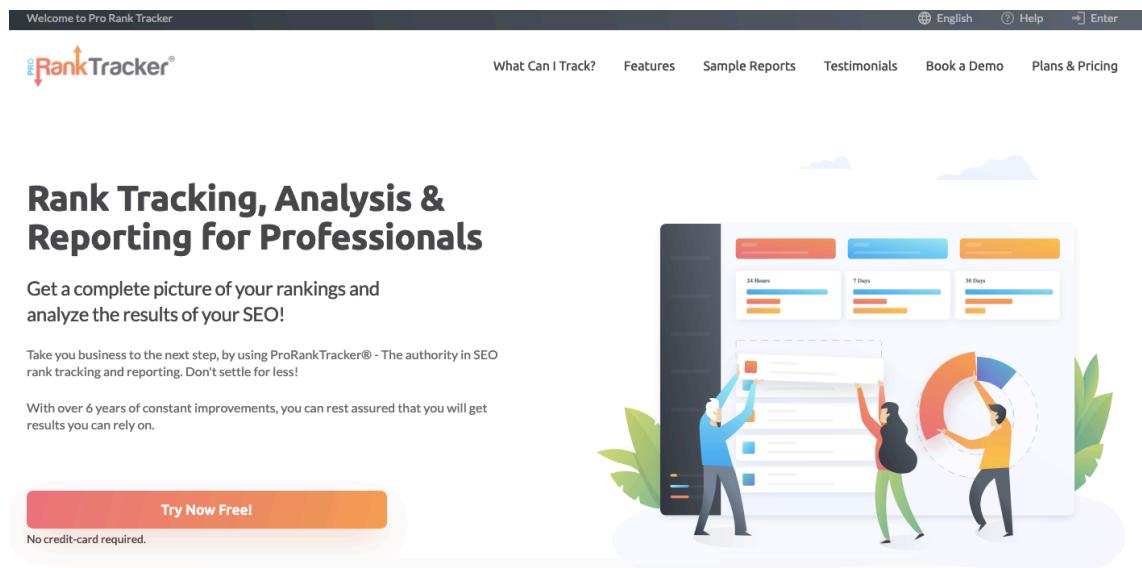


Figura 2.4.6. Página web Proranktracker

#### 2.4.1. Tabla comparativa

A continuación, se muestra una tabla resumen comparando las alternativas actuales similares a Serptracker:

Nombre	Web	Precio/mes	Palabras clave máximas	Trial
Serprobot	<a href="https://www.serprobot.com/">https://www.serprobot.com/</a>	10\$	300	Sí
Seranking	<a href="https://seranking.com/es/">https://seranking.com/es/</a>	29€	250	Sí
Accuranker	<a href="https://www.accuranker.com/">https://www.accuranker.com/</a>	49\$	500	Sí
Serpstate	<a href="https://serpstat.com/">https://serpstat.com/</a>	69\$	500	Sí
SerpWatcher	<a href="https://serpwatcher.com/">https://serpwatcher.com/</a>	49\$	200	Sí
Serptracker	<a href="https://serptracker.es/">https://serptracker.es/</a>	0€	Sin límite	.

## 2.5. ¿Por qué Ruby y NodeJS?

La decisión de elegir estos dos lenguajes se basa en gran parte en la cantidad de librerías y comunidad que poseen. En una primera fase inicial, se pensó en desarrollar el *scraper* de *proxys*, el *scraper* de resultados de búsqueda y la web que mostrase todos los resultados en NodeJS debido a que así no era necesario indagar en otros lenguajes y, con aprender uno, habría sido suficiente.

No obstante, como lenguaje de scripting NodeJS no era lo suficientemente rápido como Ruby, pero en el apartado de diseño web si que destacaba debido a su patrón MVC y librerías.

NodeJS permite consumir datos en tiempo real con una rapidez que destaca frente al resto de *frameworks* de la competencia.

Aunque se trate de un *framework* orientado al *back-end*, en los últimos ha evolucionado también orientándose en gran medida al entorno *front-end*, permitiendo así unificar en un solo framework ambas partes y conectarlas de manera sencilla.

Se ha elegido Ruby como lenguaje de scripting para realizar el scraping de los proxys y las búsquedas en Google porque cuenta con numerosos recursos para poder realizar peticiones, tales como *user-agents* [11] personalizados o *proxys* propios. La librería utilizada para el *web scraping* ha sido “*Nokogiri*” [3].

Además, posee una comunidad activa muy grande y más de 60.000 librerías para ayudar a los programadores a cumplir sus objetivos. En esta ocasión, se han utilizado librerías para conectar con la base de datos y almacenar los resultados, para realizar las peticiones y para limpiar los resultados en HTML, entre otras.

## 2.6. ¿Qué es el web scraping?

El *web scraping* se define como la técnica que se usa para extraer información de páginas web de manera automatizada. La técnica consiste en simular que una persona está navegando normalmente por la web, cuando realmente es un *bot* (programa automatizado) programado que se ha conectado mediante una conexión HTTP.

Se utiliza mucho para auditorías SEO, pero también para otros campos. Por ejemplo, en el mundo del *e-commerce* para realizar un análisis rápido y masivo de los productos de la competencia.

Otro ejemplo serían las páginas de comparativas de vuelos y viajes que se basan en el web scraping y análisis de los resultados obtenidos para conformar las ofertas que ofrecen a diario, y todo de manera automática.

Una herramienta muy popular dentro del SEO que trabaja mediante *Web scraping* es *Screaming Frog*, utilizada para encontrar cualquier fallo en el SEO Onpage de una web.

### 2.6.1. Librería Nokogiri

*Nokogiri* es una librería o gema para Ruby que se encarga de facilitar una serie de funciones y métodos predefinidos para hacer más rápida la tarea de web scraping. Se encarga de transformar una web en un objeto Ruby para utilizar dichas funciones.

Esta librería fue creada por AAron Patterson y Mike Dalessio y es de libre uso.

### 3. Fase de análisis y diseño

En esta parte del proyecto se expone todo el proceso de desarrollo de un proyecto software. A continuación, se van a detallar todos los requisitos y los casos de uso que deberá cumplir Serptracker.

El proceso de diseño de software que se ha seguido se ha basado en el estándar IEEE de ciclo de un proceso de desarrollo de software.

Durante esta fase, se ha utilizado el programa “StarUML” [12] para realizar diseño de software. Creación de diagramas y gráficos en *UML (Unified Modelling Language o Lenguaje Unificado de Modelado)*.

#### 3.1. Descripción general

##### 3.1.1. Perspectiva y objetivo del proyecto

La aplicación Serptracker tiene como objetivo permitir el seguimiento en Google de proyectos web a través de palabras clave asociadas a los mismos. Google ofrece un servicio de API para realizarlo, pero con un coste muy elevado. Esta aplicación lo hace a través de proxys, permitiendo realizar miles de peticiones y obtener los resultados para analizarlos posteriormente.

La aplicación está destinada tanto a agencias de marketing digital que lleven el SEO de sus clientes y quieran utilizar los resultados a modo de informe, ya sea para identificar la eficacia de sus estrategias, tanto como para entregar resultados a los clientes. También está destinada a pequeños usuarios que tengan uno o varios proyectos web y quieran tener estadísticas reales de los avances.

### 3.2. Requisitos funcionales

Siguiendo la normativa estándar *Web Content Accesibility Guidelines* RFC 2396 RFC 2396 del IETF (Grupo de Trabajo de Ingeniería de Internet) y recomendaciones establecidas por la W3C.

Requisitos relacionados con los usuarios y la finalidad de la aplicación

- **REF01.** Cualquier usuario que entre en la aplicación web tendrá que identificarse mediante un usuario y una contraseña.
- **REF02.** Si un usuario accede a la web y no posee cuenta de usuario, podrá crear una dirigiéndose al apartado de “registrar cuenta”.
- **REF03.** El registro de usuario, referenciando al **REF02**, poseerá los siguientes apartados necesarios para el registro de una cuenta:
  - Email: email del usuario, necesario para enviarle notificaciones, como una nueva contraseña, si fuese necesario.
  - Usuario: nombre de usuario necesario para acceder a la aplicación.
  - Contraseña: contraseña de usuario.
- **REF04.** Una vez iniciada la sesión, el usuario será redirigido a un panel de control (*dashboard*) donde podrá ver de un vistazo los resultados de sus últimos proyectos.
- **REF05.** El usuario podrá salir de la aplicación pinchando el botón de “cerrar sesión”.
- **REF06.** En referencia al REF05, si el usuario sale de la aplicación, deberá volver a iniciar sesión para entrar, dado que la sesión creada habrá sido destruida.
- **REF07.** La aplicación permitirá añadir nuevos proyectos en el apartado de la web llamada “nuevo proyecto”.
- **REF08.** En referencia al REF07, al añadir un proyecto será necesario añadir los siguientes datos:
  - **Nombre del proyecto:** nombre identificativo.
  - **Dirección:** URL de la página web que se usará como base para rastrear los resultados de búsqueda.
  - **Palabras clave:** son las palabras con cada una de las cuales se realizará una búsqueda. También llamadas *keywords*.

- **REF09.** Una vez añadido un nuevo proyecto, se realizará de manera automática un análisis del SEO onpage de la página principal para detectar si existen los siguientes datos, muy importantes para el SEO:
  - H1: se trata del título o encabezado principal de la web. Es el título que más debería identificar a la web.
  - H2: encabezados H2 de la web.
  - H3: encabezados H3 de la web.
  - *Meta title*: título perteneciente a los metadatos de la web. Es el título que se mostrará una vez se busque en Google, no tiene por qué ser el mismo título que el H1.
  - *Meta description*: meta descripción de la web. Es la pequeña descripción que aparece debajo del título cuando aparece un resultado en una búsqueda en Google.
  - *DoFollow*: son los enlaces salientes o internos en la web que transmiten autoridad.
  - *NoFollow*: son los enlaces salientes o internos de la web que no transmiten autoridad.
- **REF10.** La web debe tener un apartado de Aviso Legal donde se informe a los usuarios de la página de que la web cumple con las leyes vigentes.
- **REF11.** Debe existir un apartado que contenga los términos y condiciones de uso.
- **REF12.** La web tendrá un apartado donde se informe a cerca de la Política de privacidad. En ella, se mostrará cómo la web se acoge a la Ley Orgánica 3/2018 del 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales (LOPDGDD), que desarrolla el Reglamento (UE) 2016/679 del Parlamento Europeo y del consejo de 27 de abril de 2016 en relación a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos, y que regula cualquier actividad donde se recopilen datos personales de primer nivel.
- **REF13.** La web no requerirá nada adicional para la ejecución de los proyectos.
- **REF14.** La web será multiplataforma, permitiendo acceder a la misma desde cualquier navegador y desde cualquier sistema operativo, incluyendo dispositivos móviles.
- **REF15.** El administrador del sitio podrá dar de alta o baja usuarios de manera manual.
- **REF16.** El administrador podrá eliminar, actualizar o añadir proyectos a usuarios de manera manual.
- **REF17.** La página web tendrá un formulario de contacto para los usuarios que deseen enviar alguna duda o sugerencia al administrador de la página.

### **3.3. Requisitos no funcionales**

Los requisitos no funcionales hacen referencia a la parte servidor, incluyendo todo lo referente a seguridad, integridad y disponibilidad.

**RNF01.** La aplicación tendrá un diseño responsive que se adapte a todo tipo de dispositivos.

**RNF02.** La aplicación será multiplataforma al tratarse de una aplicación web.

**RNF03.** La aplicación estará hecha siguiendo el patrón MVC para obtener todas las ventajas que ofrece.

**RNF04.** El sitio web tendrá una base de datos MySQL.

**RNF05.** La aplicación estará protegida ante cualquier tipo de ataque, incluyendo inyecciones SQL.

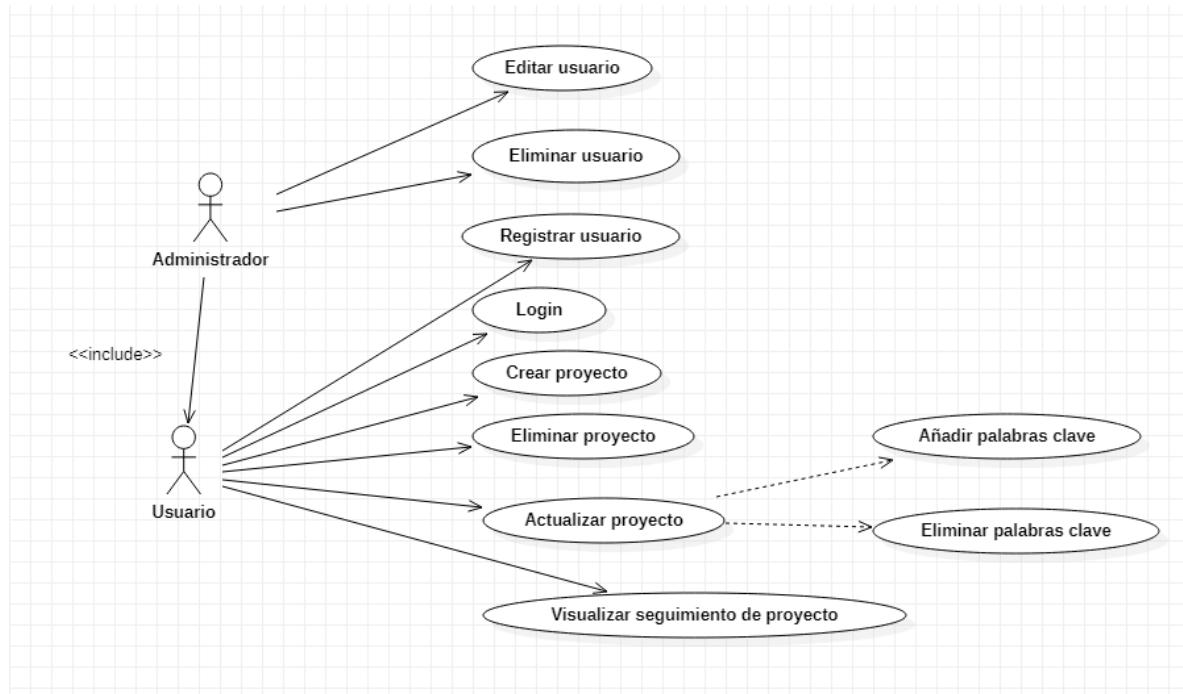
**RNF06.** El sitio web tendrá un sistema de rutas que facilite la programación de este.

### 3.4. Casos de uso

Los casos de uso se utilizan para representar el comportamiento de una herramienta o sistema cuando interviene un usuario en el mismo. Se representa con actores y estados.

En este caso concreto, Serptracker posee dos tipos de actores:

- **Administrador:** el administrador puede eliminar, añadir o actualizar cualquier proyecto de un usuario. También tiene la posibilidad de crear los suyos propios y modificar o eliminar usuarios de la herramienta.
- **Usuario:** engloba cualquier persona registrada en la aplicación. Pueden crear proyectos, añadir palabras clave a los mismos y visualizar resultados obtenidos. También pueden modificar proyectos ya creados, pero siempre solo los suyos propios.



### 3.5. Diagrama entidad relación

La base de datos almacena los datos de los usuarios en una tabla llamada “*login*”. En la tabla denominada “*position*” se almacenarán las posiciones de una palabra clave, así como la fecha y el proyecto al que pertenece.

La tabla “*project*” es la encargada de guardar todos los proyectos de los usuarios.

La tabla “*proxys*” almacena los proxys, tanto chequeados como sin chequear. Esta tabla se borra y actualiza cada día debido al script de obtención de nuevos proxys mediante web scraping.

La tabla “*seonpage*” almacena el análisis de la web de los elementos que afectan al seo interno de la página.

La tabla “*keyword*” contiene todas las palabras clave y los proyectos a los que pertenecen.

Todas las tablas se relacionan entre sí mediante claves ajenas.

El diagrama E-R es el siguiente:

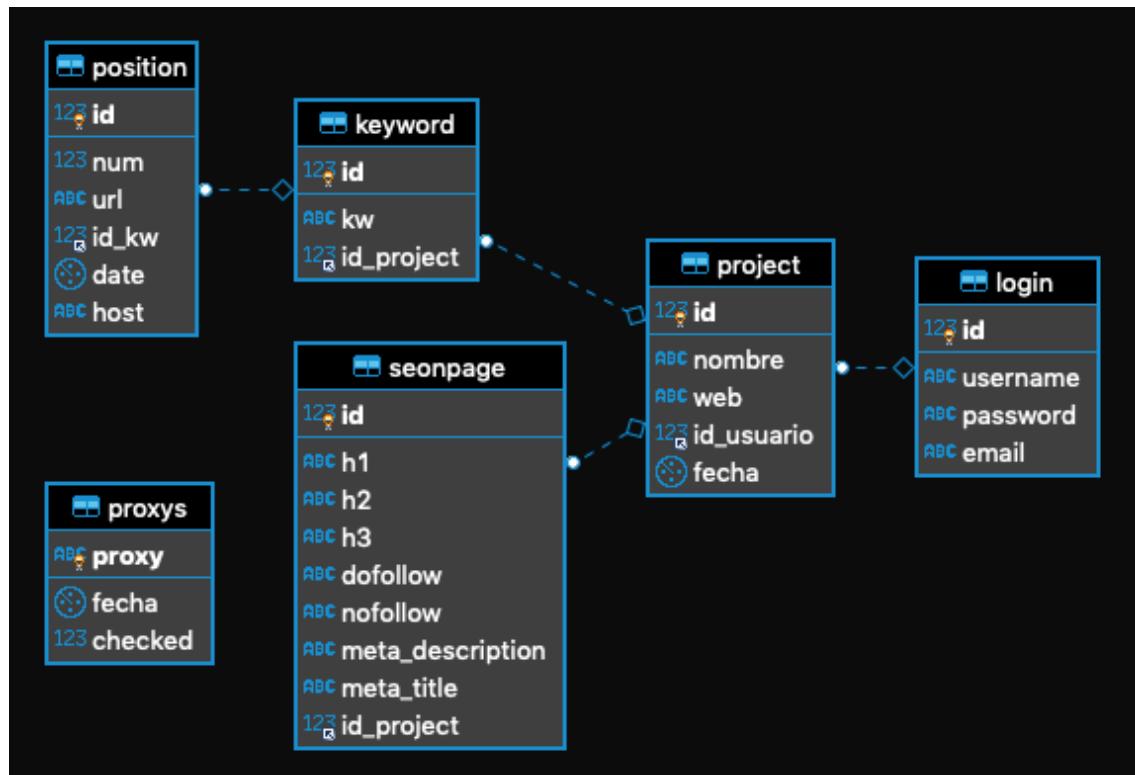


Figura 3.5.1. Diagrama entidad-relación

### 3.6. Diagrama de actividad

#### 3.6.1. Obtención de proxys

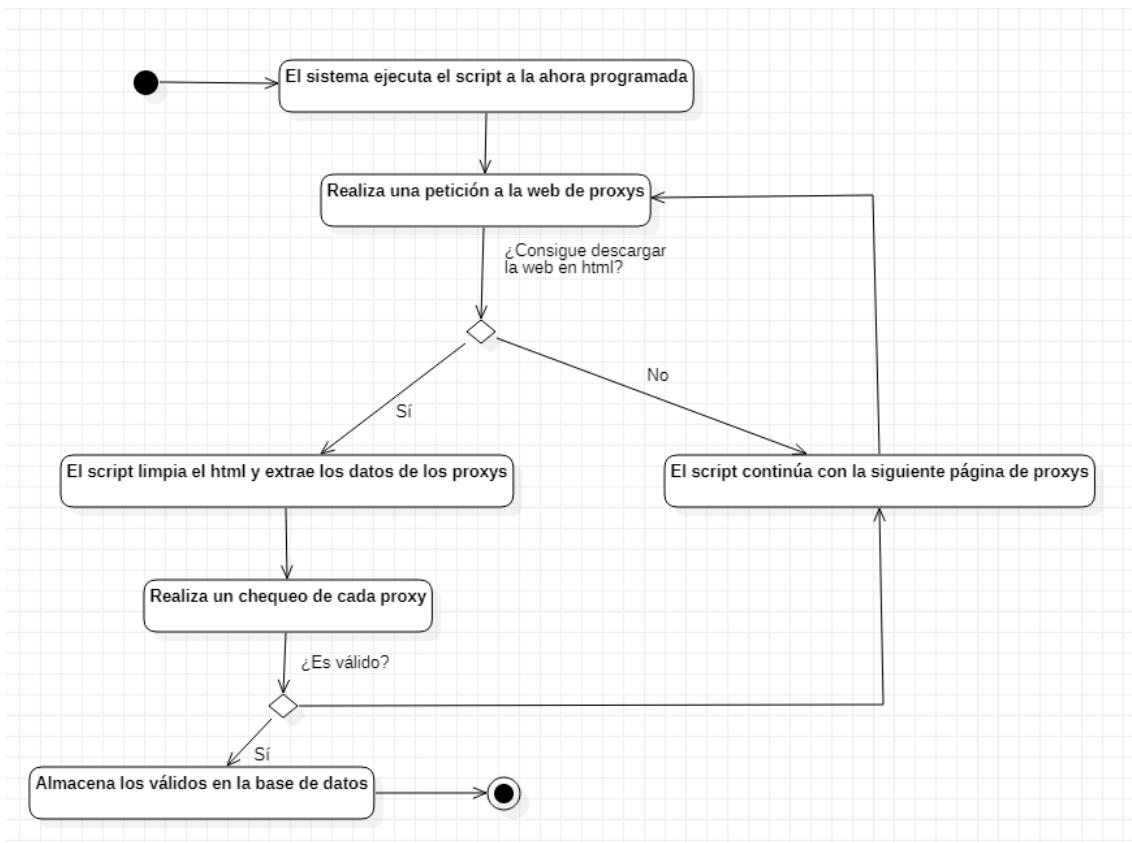


Figura 3.6.1.1. Diagrama de actividad que representa la obtención de datos de los *proxys* a utilizar.

### 3.6.2. Obtención de posiciones

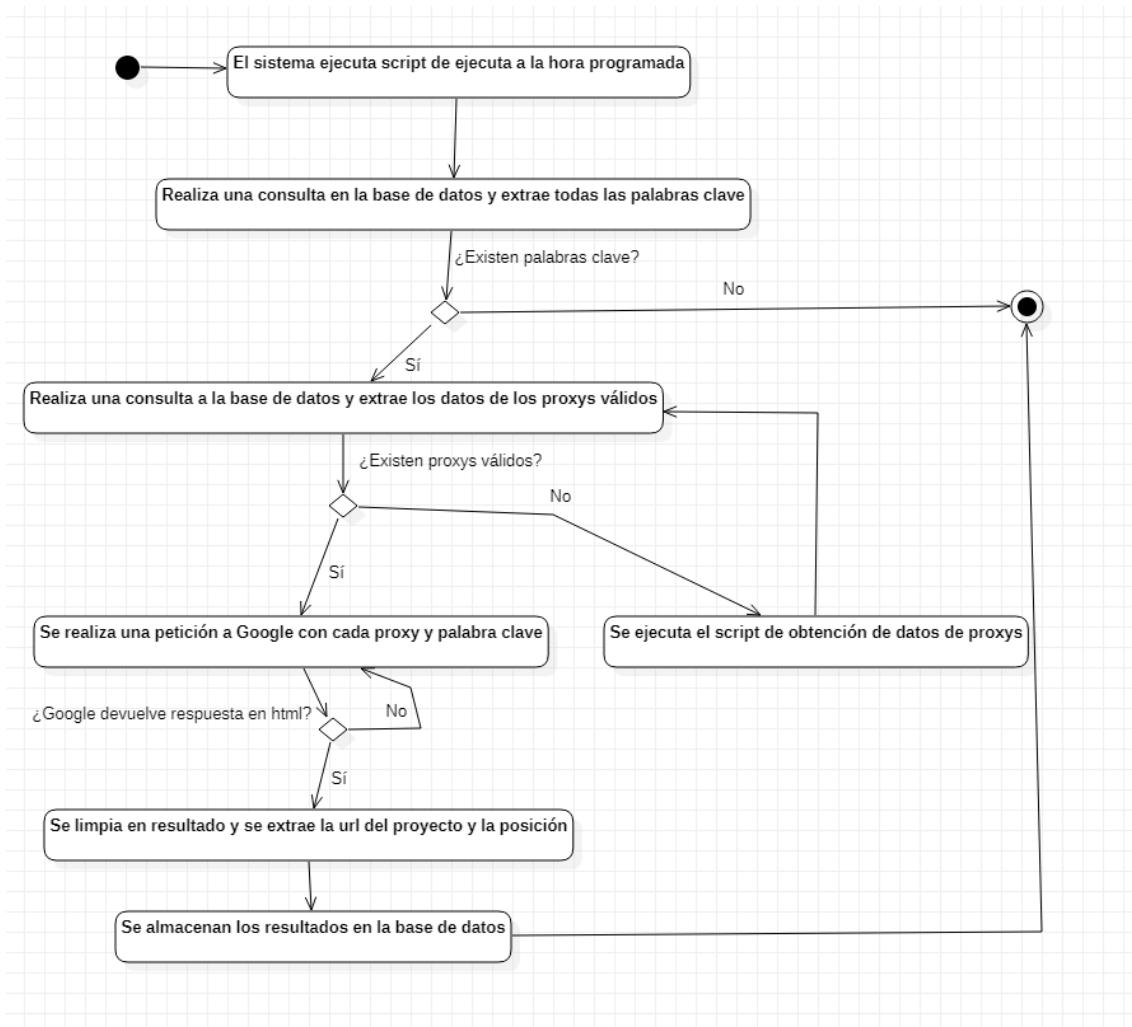


Figura 3.6.2.1. Diagrama de actividad que representa el proceso de obtención y procesado de los resultados de búsqueda en Google.

### 3.7. Diseño de prototipos de baja fidelidad

Para el diseño de los prototipos de baja fidelidad se ha utilizado el programa “*Balsamic Mockups*”. Se trata de una herramienta para diseñar de forma fácil maquetas de aplicaciones, tanto webs como aplicaciones móviles.

La página de inicio del sitio web Serptracker consta de un resumen de las posiciones de los proyectos añadidos. También llamada “*Dashboard*”.

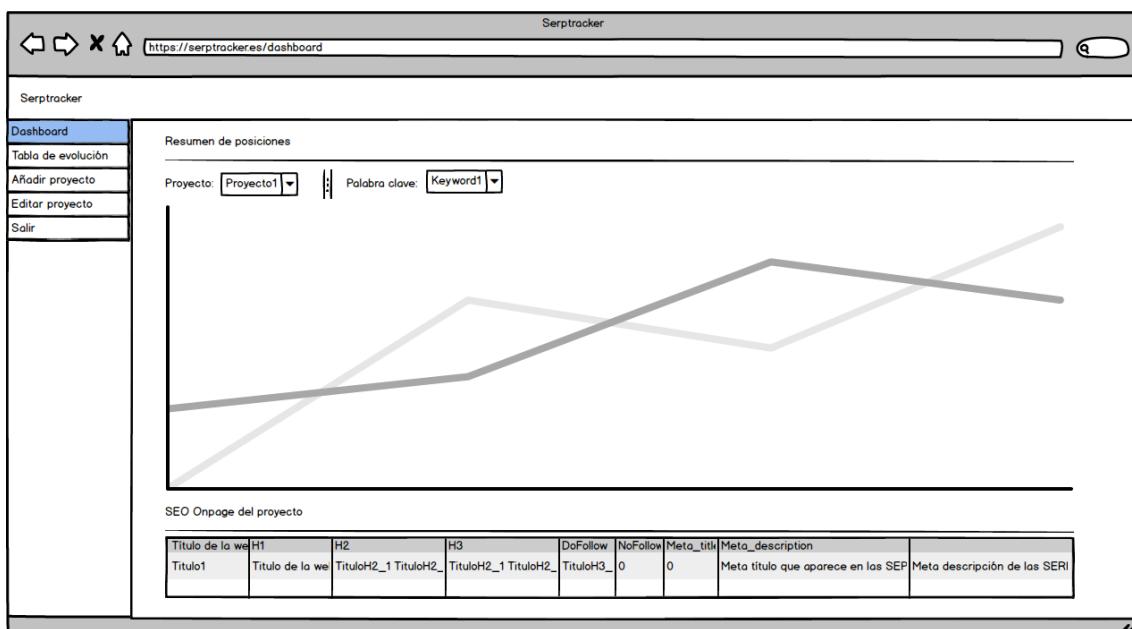


Figura 3.7.1. Inicio de la web donde se muestra un resumen de los proyectos.

La página para añadir un nuevo proyecto solicita los siguientes datos:

- Nombre del proyecto.
- URL del sitio web.
- *Keywords*: son las palabras clave de las que se realizará el seguimiento.

The screenshot shows a web browser window for 'Serptracker' at the URL <https://serptrackeres/new>. On the left, a vertical sidebar menu includes 'Añadir proyecto' (selected), 'Editar proyecto', 'Tabla de evolución', 'Dashboard', and 'Salir'. The main content area has a title 'Añadir nuevo proyecto'. It contains three input fields: 'Nombre proyecto:' with a placeholder box, 'Url del sitio web:' with a placeholder box, and a larger text area for 'Keywords (1 por linea:)'. At the bottom is a 'Crear proyecto' button.

Figura 3.7.2. Apartado web donde se añaden nuevos proyectos

El apartado de la web destinado a editar un proyecto ya existente permite añadir o eliminar palabras clave.

The screenshot shows a web browser window for 'Serptracker' at the URL <https://serptrackeres/edit>. The sidebar menu is identical to Figure 3.7.2. The main content area has a title 'Editar proyecto' and a 'Keywords (1 por linea:)' text area. To the right is a table listing keywords: Kw1, Kw2, Kw3, and Kw4. Each keyword has a small circular icon with a minus sign to its right, indicating it can be deleted. At the bottom is an 'Añadir keywords' button.

Figura 3.7.3. Apartado web donde se añaden nuevas palabras clave a un proyecto existente.

La página de acceso, también llamada “**login**”, consta de un formulario en el que se solicita usuario y contraseña para acceder. También posee la opción de registrar una nueva cuenta de usuario.

The screenshot shows a web browser window for 'Serptracker' at the URL <https://serptrackeres/>. The page features a central login form with the following fields:

- A header labeled "Serptracker".
- A "Usuario:" label with an associated input field.
- A "Contraseña:" label with an associated input field.
- A "Entrar" button below the password field.
- A "Registrar cuenta" link at the bottom of the form.

Figura 3.7.4. *Login* de usuario.

La página destinada al registro de un nuevo usuario solicita los siguientes datos:

- Nombre de usuario.
- Email.
- Contraseña de usuario.

The screenshot shows a web browser window for 'Serptracker' at the URL <https://serptrackeres/register>. The page displays a registration form titled "Crear nueva cuenta" with the following fields:

- A header labeled "Crear nueva cuenta".
- A "Usuario:" label with an associated input field.
- An "Email:" label with an associated input field.
- A "Contraseña:" label with an associated input field.
- A "Crear cuenta" button at the bottom of the form.

Figura 3.7.5. Formulario de registro de usuario.

La página encargada de mostrar la evolución de posiciones permite visualizar el número exacto de posiciones ganadas o perdidas y la palabra clave asociada.

The screenshot shows a web browser window for 'Serptracker' at the URL <https://serptrackeres/evolution>. On the left, there's a sidebar with links: 'Tabla de evolución' (selected), 'Dashboard', 'Añadir proyecto', 'Editar proyecto', and 'Salir'. The main content area has a title 'Tabla de evolución de posiciones' and a dropdown 'Proyecto' set to 'Proyecto1'. A table lists 10 keywords, each with its current position, change from yesterday, change over 7 days, and change over 30 days. All keywords show a gain of 8 positions.

Keyword	Posición actual	Posición respecto ayer	Posición respecto a los últimos 7 días	Posición respecto a los últimos 30 días
Keyword1	4	4(-1)	7(+3)	12(+8)
Keyword2	4	4(-1)	7(+3)	12(+8)
Keyword3	4	4(-1)	7(+3)	12(+8)
Keyword4	4	4(-1)	7(+3)	12(+8)
Keyword5	4	4(-1)	7(+3)	12(+8)
Keyword6	4	4(-1)	7(+3)	12(+8)
Keyword7	4	4(-1)	7(+3)	12(+8)
Keyword8	4	4(-1)	7(+3)	12(+8)
Keyword9	4	4(-1)	7(+3)	12(+8)
Keyword10	4	4(-1)	7(+3)	12(+8)

Figura 3.7.6. Apartado web con un resumen de evolución de posiciones indicando el número ganado o perdido.

## 4. Fase de desarrollo

Esta etapa corresponde a la implementación de la herramienta **Serptracker**. Se basa en la fase de análisis y diseño, y se ha realizado teniendo en cuenta tanto los requisitos funcionales como los no funcionales y los casos de uso.

Para el diseño se ha seguido la normativa ISO/IEC 25010 referente a la evaluación de la calidad de un producto software [5].

La calidad de un software se interpreta como el grado en que el software satisface los requisitos del usuario y organiza la calidad del software en características y sus características. Tiene ocho características generales, representadas por la siguiente imagen:



Figura 4.1. Imagen representativa de las características de un software de calidad según la normativa ISO 25000.

### 4.1. Análisis del funcionamiento del motor de búsqueda Google

Google es un motor de búsqueda que posee diferentes etiquetas en su URL, contenido de esta manera los parámetros y valores a utilizar. La información para extraer para el análisis debía constar de la posición en la que se encontraba la web de un proyecto para determinada palabra clave.

En este ejemplo de búsqueda se puede observar como el parámetro de la URL llamado “**num**” indica el número de resultados, el “**client**” indica el navegador web desde el que se realiza la búsqueda y “**q**” se refiere a la búsqueda (**query**):

```
https://www.google.es/search?num=100&client=firefox-b-d&q=${keyword}
```

Con estos datos y realizando *web scraping* se debían conseguir los datos, pero la complejidad fue mayor, dado que había que realizar peticiones simulando ser una persona y mediante diferentes IPs, puesto que si no saltaba un *reCaptcha*.

#### 4.2. Búsqueda de solución para realizar una consulta de manera automática

Para el análisis de cómo ejecutar una búsqueda automática en Google se hizo primero una búsqueda sin cabeceras en la petición desde un script programado en Ruby y se detectó que Google no devolvía resultados.

Además, la aplicación requería numerosas búsquedas simultáneas, no solo una, por lo que era necesaria una solución.

Se realizó un script con Ruby que realizaba la petición con diferentes *user-agents* [11] (parte de una petición a un servidor web donde se indica el dispositivo utilizado para conectarse, su sistema operativo y la versión del navegador web) simulando así ser un humano. Esto no fue suficiente, puesto que cuando se realizaban varias búsquedas seguidas saltaba un *reCaptcha* que había que llenar manualmente para poder continuar.



Figura 4.2.1. *ReCaptcha* de Google.

La solución encontrada fue la utilización de diferentes *proxys* en cada búsqueda automática.

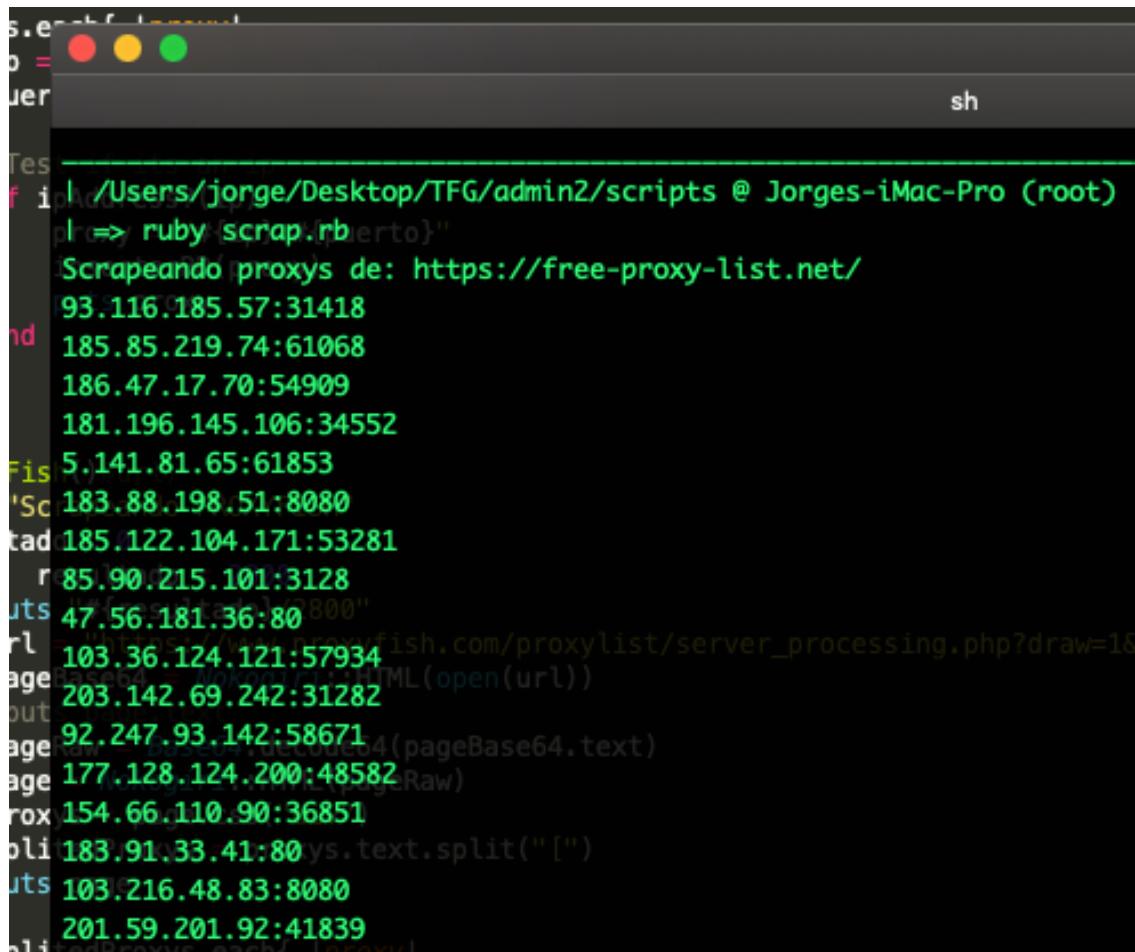
#### 4.3. Obtención de datos de proxys de manera masiva

Para la obtención de estos *proxys*, y su posterior chequeo antes de usarlos, se realizó otro *script*, también programado en Ruby.

Este *script* es el encargado de conseguir miles de datos de conexión a *proxys* mediante *web scraping* a diferentes páginas escogidas entre las más populares. Los datos que se obtienen son la dirección IP y el puerto para acceder a través del *proxy*. Este método de obtención de información de manera masiva también se denomina modo *bulk*.

Utilizando peticiones HTML [10] a las webs se obtienen los datos de los *proxys* en HTML y, mediante la librería *Nokogiri* de Ruby se limpia todo ese HTML obtenido para obtener solo IP y puerto de cada proxy.

Posteriormente se almacenan en una base de datos MySQL de manera masiva y automática.



```
s.e@Jorges-iMac-Pro: ~
```

```
jorge: ~
```

```
user
```

```
sh
```

```
Test
```

```
file /Users/jorge/Desktop/TFG/admin2/scripts @ Jorges-iMac-Pro (root)
```

```
[ -> ruby scrap.rb ]
```

```
Scrapeando proxys de: https://free-proxy-list.net/
```

```
93.116.185.57:31418
```

```
185.85.219.74:61068
```

```
186.47.17.70:54909
```

```
181.196.145.106:34552
```

```
5.141.81.65:61853
```

```
183.88.198.51:8080
```

```
185.122.104.171:53281
```

```
85.90.215.101:3128
```

```
47.56.181.36:80
```

```
103.36.124.121:57934
```

```
203.142.69.242:31282
```

```
92.247.93.142:58671
```

```
177.128.124.200:48582
```

```
154.66.110.90:36851
```

```
183.91.33.41:80
```

```
103.216.48.83:8080
```

```
201.59.201.92:41839
```

```
103.216.48.83:8080
```

```
201.59.201.92:41839
```

Figura 4.3.1. Ejecución del *script* de obtención de los datos de *proxys* mediante *web scraping*.

#### 4.4. Verificación de proxys válidos

La verificación de *proxys* se realiza posteriormente al guardado de los mismos en la base de datos.

El chequeo consiste en realizar una petición a una web cualquiera y ver el código de respuesta de esta, si devuelve una cabecera válida, se marca como válido en la base de datos para su posterior utilización en las consultas a Google.

#### 4.5. Almacenamiento de resultados en base de datos MySQL

Todos los datos de la aplicación, tanto los datos de los *proxys* como los datos de usuarios y proyectos son almacenados mediante consultas con sintaxis SQL.

Según el lenguaje utilizado (Javascript o Ruby) se utilizan unas librerías u otras, pero siempre mediante la sintaxis original de SQL.

Javascript se utiliza en la web de Serptracker desarrollada con NodeJS.

#### 4.6. Desarrollo de aplicación con NodeJS

Para el desarrollo con NodeJS se ha utilizado una arquitectura MVC.

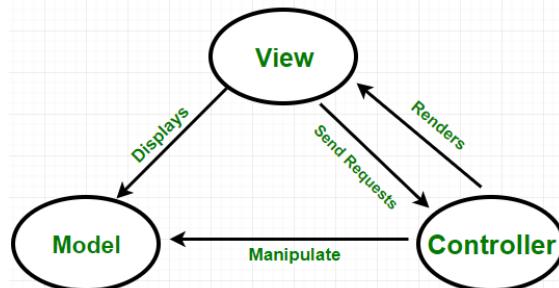


Figura 4.6.1. Esquema Modelo Vista Controlador.

Para facilitar más la tarea de programación y hacerla más ágil, se han usado librerías como **express**, la cual permite usar mecanismos para escritura de manejadores de peticiones HTTP con diferentes rutas. También permite renderizar vistas de manera sencilla, basándose en plantillas EJS.

La estructura final del proyecto quedó de la siguiente forma:

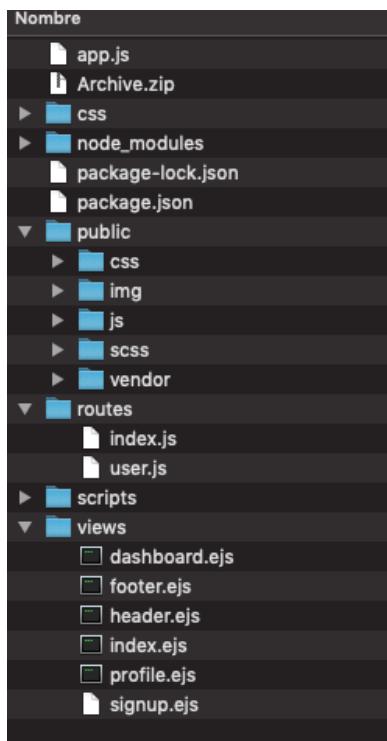


Figura 4.6.2. Estructura de carpetas del proyecto con NodeJS.

#### 4.7. Elección del servidor donde alojar Serptracker

En un principio se iba a utilizar una Raspberry pi 3, no obstante, y dado que NodeJS necesitaba de gran memoria para su ejecución y los scripts de *web scraping* utilizaban múltiples hilos, se reemplazó por una placa de servidor “Supermicro X8STE” junto con un procesador Xeon “X5650”.

La placa está preparada para tener hasta 24Gb de memoria RAM, por lo que es suficiente para la ejecución de cualquier tarea necesaria.

No obstante, las cajas de servidor son de 1U (1 unidad de altura):



Figura 4.7.1. Servidor de altura 1U.

La placa comprada para el proyecto venía sin caja y sin fuente de alimentación, por lo que hubo que adaptar una fuente de alimentación de ordenador y diseñar unas piezas en 3D para realizarle una estructura al servidor junto con tubos de PVC para fontanería:

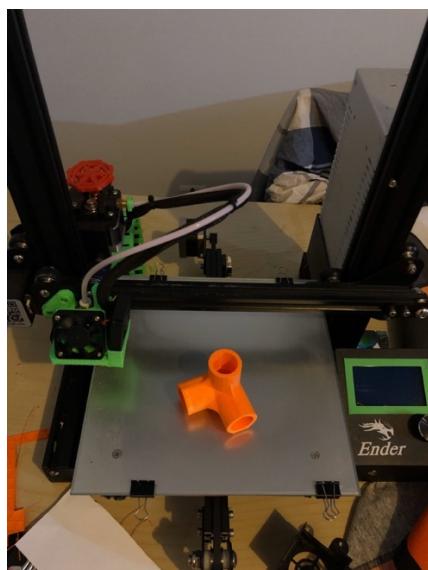


Figura 4.7.2. Codera diseñada e imprimida en impresora 3D.

La caja se conformó uniendo los tubos de PVC mediante las coderas diseñadas:



Figura 4.7.3. Estructura construida con el servidor ya instalado.

#### 4.8. Configuración del servidor y ejecución de Serptracker

El servidor se trata de un Ubuntu Server configurado con NodeJS ejecutándose continuamente.

Los *scripts* de Ruby que realizan el web scraping de los *proxys* y las búsquedas en Google están programados para ejecutarse a las 00:00 diariamente.

Mediante *Crontab* (administrador que permite ejecutar procesos automatizados en segundo plano) se nos permite ejecutar tareas de manera programada.

#### 4.9. Configuración del dominio

Se compró un dominio en <https://DonDominio.com/> llamado <https://serptracker.es/> para el proyecto.

En el panel de configuración del dominio hay que realizar una redirección a la IP del servidor anteriormente instalado y se mostrará la aplicación.

## 5. Fase de pruebas: caso práctico

### 5.1. Introducción

Como modo de prueba, se añadió un proyecto propio a la herramienta con el fin de observar si los resultados que se iban obteniendo eran fieles a la realidad.

### 5.2. Creación del proyecto

A continuación, se explica el desarrollo llevado a cabo desde la creación del proyecto hasta el análisis posterior de las posiciones obtenidas. Este análisis de posiciones se realiza mediante la herramienta Serptracker en todo momento.

#### 5.2.1. Elección del nicho

A la hora de elegir un proyecto destinado a la monetización es muy importante desde un primer momento la elección correcta de la temática. Las palabras clave o *keywords* de ese nicho deben tener las búsquedas suficientes como para que el RPM resulte rentable.

Como el nicho escogido está orientado a monetizar mediante Amazon afiliados y Google Adsense (plataforma para anunciantes de Google), debemos fijarnos en las comisiones por venta en el país para el que se oriente la web, en este caso se eligió un nicho de “máquinas de pintura *airless*”, ya que era un nicho de competencia media y de la temática de bricolaje, la cual da un 8% de comisión por venta en Amazon España.

#### 5. Tarifas de Comisiones Estándar del Programa para el Sitio de Amazon ES

Categoría de productos	Tarifas fijas de las comisiones estándar del programa	Condiciones especiales
Amazon Moda:		<b>Compras adscritas directas:</b>
Ropa y accesorios, zapatos, bolsos, carteras, relojes, equipaje, joyería y las marcas privadas de Amazon moda (mujer, hombre, niños)	11%	Ingresos adscritos por mes = menos de 2500 €
	12%	Ingresos adscritos por mes = 2500 € o más
	1,5%	<b>Compras adscritas indirectas</b>
Handmade	10%	
Aparatos de cuidado personal, salud y cuidado personal, oficina y papelería, bebé, libros, libros electrónicos de Kindle, coche y moto, belleza, productos para mascotas	7%	
Hogar:		<b>Compras adscritas directas:</b>
Muebles, bricolaje, hogar, cocina y comedor, patio, césped y jardín, herramientas eléctricas y manuales	7%	Ingresos adscritos por mes = menos de 40 000 €
	8%	Ingresos adscritos por mes = 40 000 € o más
	1,5%	<b>Compras adscritas indirectas</b>

Figura 5.2.1.1. Resumen de porcentajes de comisión según la categoría en Amazon Afiliados de España.

#### 5.2.2. Keyword research

Este término se refiere a la búsqueda de palabras clave con volúmenes de búsqueda. Estas palabras son las que intentaremos posicionar en Google y conseguir así visibilidad y, por lo tanto, ventas y comisiones.

Esta búsqueda de palabras clave se realizó utilizando el planificador de palabras de Google, el cual permite obtener palabras clave relacionadas a partir de una principal, en este caso “máquinas de pintura”. También permite obtener volúmenes de búsqueda exactos si tienes una campaña de adwords activa.

### 5.2.3. Instalación y configuración de la web

La web se instaló en un servidor dedicado propio, creado a partir de una placa de servidor y un procesador Xeon. Este mismo servidor también aloja la herramienta desarrollada.

En el servidor web creado se instaló como CMS (gestor de contenido) Wordpress.

Se compró el dominio <https://maquinasdepintura.com/> para realizar una redirección de este al servidor y visualizar la web instalada.

Para la mayor parte del contenido se contrató una empresa llamada *Buyatext* (<https://buyatext.com/> ).

### 5.2.4. Indexación en Google

La indexación en Google se refiere a la publicación de la web para que sea encontrada por los motores de búsqueda, en este caso y a través de la herramienta “*Search Console*” de Google se forzó la indexación de las urls creadas.

## 5.3. Seguimiento de resultados con Serptracker

Una vez el proyecto de “Máquinas de pintura” fue publicado en Google, se creó un proyecto en Serptracker, añadiendo un nombre de proyecto, las palabras clave a rastrear y la URL de la página web.

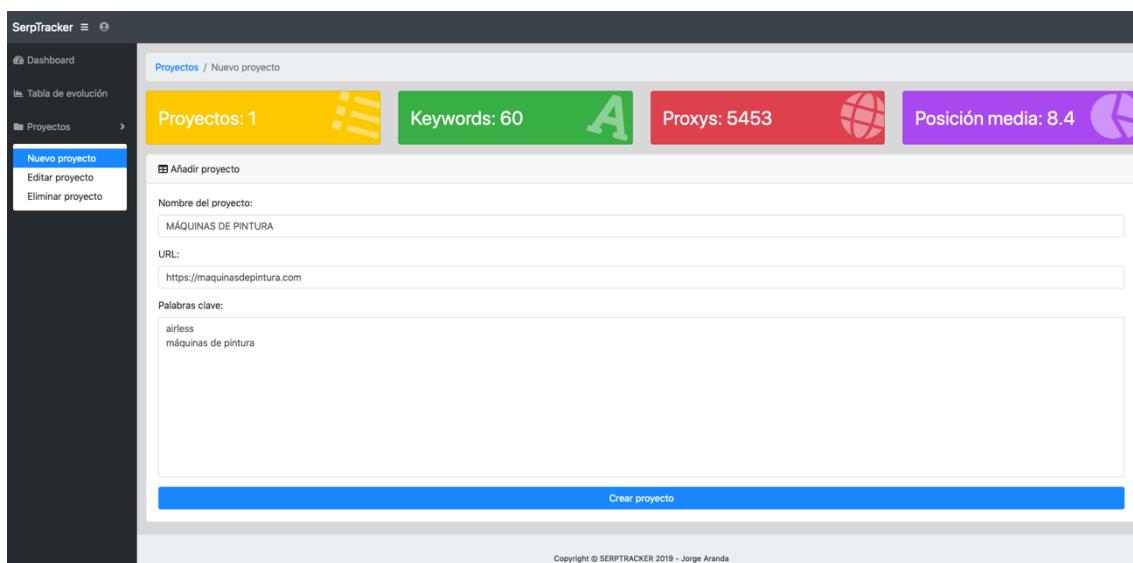
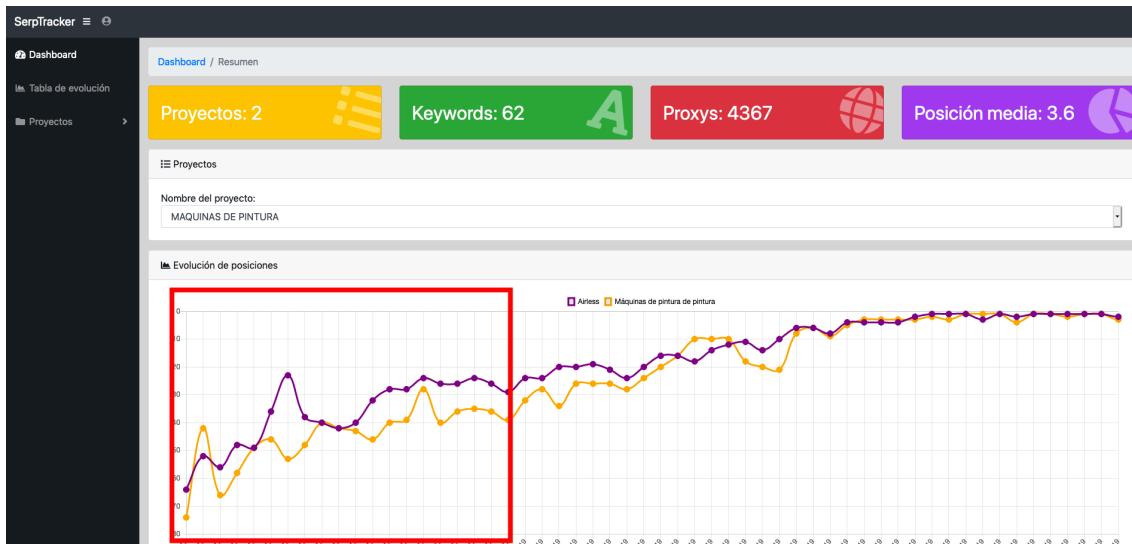


Figura 5.3.1. Formulario para añadir un proyecto nuevo.

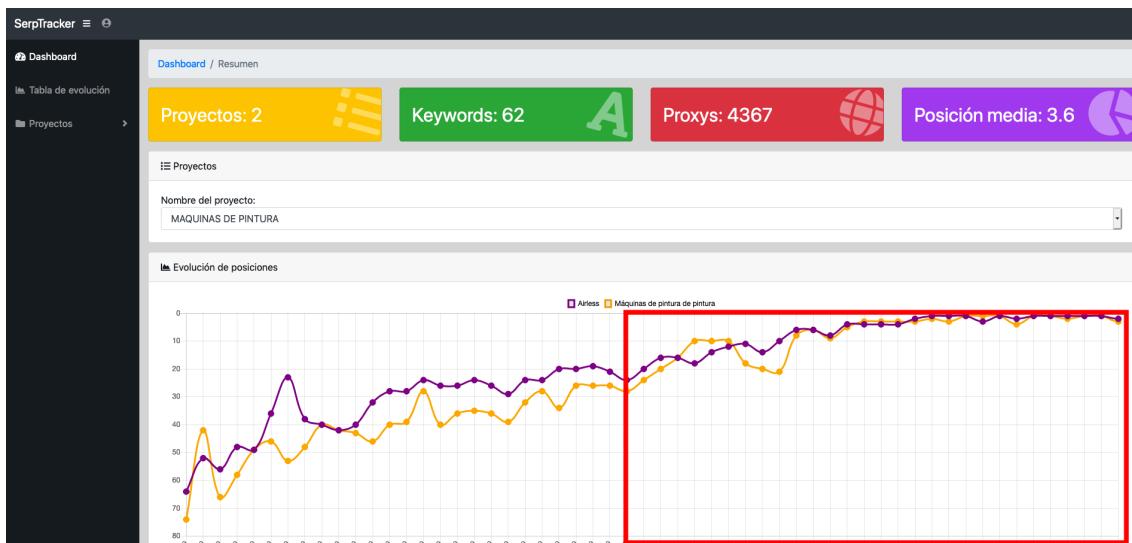
Una vez añadido, la herramienta realiza la búsqueda de cada una de las palabras clave añadidas para comenzar con el seguimiento. También realiza un análisis diario del SEO Onpage de la web para detectar si hay algún fallo en los encabezados o enlaces.

A lo largo de las semanas, comenzó a escalar posiciones en Google debido al contenido creado en la página.

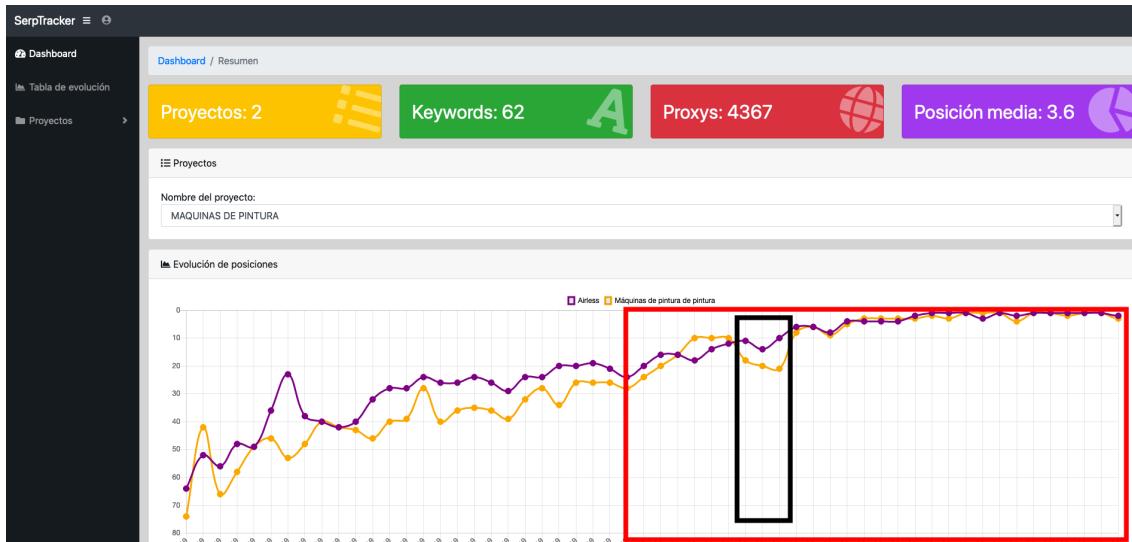


También se empezaron a crear enlaces desde otras webs (*link building*) para que el dominio obtuviese más autoridad y escalase más posiciones en Google. Estos enlaces se compraron desde artículos de periódicos relacionados con el bricolaje. Los enlaces se compraron en <https://prensarank.com/>.

En Serptracker se observa una clara subida cuando fueron publicados estos enlaces. La subida fue gradual debido a que Google procesa la autoridad que se transmite de manera progresiva.



No obstante, se detectó una bajada de posiciones brusca tras la publicación de una serie de enlaces en varios perfiles de foros de temática general.



La solución consistió en *desindexar* (eliminar dichos enlaces creados de Google) los enlaces, de esta forma las posiciones volvieron a tener una tendencia alcista.

La web se empezó monetizando con Amazon afiliados y se terminó añadiendo también Google Adsense.

#### 5.4. Conclusiones de la prueba en Serptracker

La prueba en Serptracker se realizó desde un proyecto iniciado desde 0, por lo que era necesario ir ganando posiciones y registrarlas en la herramienta.

Ha cumplido las expectativas ampliamente, ya que se ha ido notando a lo largo de las semanas un aumento de las visitas directamente proporcional al aumento de las posiciones recogidas en Serptracker.



Figura 5.4.1. Captura de Google Analytics donde se observa que la subida de las visitas a la web es gradual y coincide con la subida de posiciones en el buscador registradas en Serptracker.

La aplicación detectó una bajada brusca de posiciones justo en el momento en el que se comenzó a enlazar esta nueva web desde otras. Una vez detectada la bajada, se eliminó el enlace y las posiciones volvieron a subir.

Los resultados del seguimiento de posiciones que recoge y muestra Serptracker se ajustan a la realidad, debido a que los resultados que recoge son idénticos a los que recogería un usuario haciendo una búsqueda en Google.

Gracias a este seguimiento, se han ido tomando una serie de decisiones en la web de prueba creada que la han llevado hasta las primeras posiciones del buscador. Esto se ha traducido en un aumento de la visibilidad y de las ventas.

## 6. Accesibilidad y Usabilidad

### 6.1. Accesibilidad

La accesibilidad se utiliza como referencia al uso que podrían realizar las personas con algún tipo de discapacidad en un sitio web, de manera que puedan interactuar y entender todo lo referente a la página.

Este estudio se refiere a la accesibilidad de la aplicación web Serptracker, basándolo en las pautas y guías de la normativa de Accesibilidad para el Contenido Web 2.0 (WCAG). Dicha guía contiene las recomendaciones más importantes para considerar un sitio web accesible:

- **Perceptible:** toda la información mostrada en la web debe ser percibida sin ningún tipo de dificultad, incluyendo un fácil acceso mediante los componentes que posea la web.
- **Operable:** será operable todo contenido de la web.
- **Comprensible:** la información mostrada será comprensible.
- **Robusto:** contenido robusto para ser interpretado sin ningún tipo de duda.

### 6.2. Usabilidad

La usabilidad de un sitio web se define como la facilidad de uso que tiene un usuario para navegar por la página. Una página usable será aquella que permita una navegación sencilla, segura, intuitiva y amigable.

Utilizando el estado del arte y el estudio de la competencia se han encontrado una serie de características comunes que hacen a un sitio web usable:

- **Página web intuitiva.**
- **Carga rápida** de los elementos e información de la web.

- **Reducción y eliminación de errores** del sitio que dificulten la correcta usabilidad de la herramienta.
- **Curva de aprendizaje sencilla**, ofreciendo manuales de usuario adaptados a las necesidades de los usuarios.

Para lograr una usabilidad óptima se han valorado distintos apartados de usabilidad web:

- **Contenido**: representación lo más visual y sencilla posible de los datos tras un análisis de los datos del proyecto elegido.
- **Diseño responsive**: la aplicación web está realizada incluye la librería bootstrap [7], cuyos elementos contienen ya las características para adaptarse a cualquier tipo de dispositivo, ya sea móvil o escritorio.
- **Página web intuitiva**: todo el contenido está colocado para tener un fácil acceso al mismo.
- **Menús**: menús sencillos, cortos y con el contenido claro.
- **Colores del diseño**: colores elegidos para una agradable lectura de las tablas, los formularios y los gráficos.

## 7. Licencia Documental

Copyright (C) 2019 Jorge Aranda Barbero.

Se otorga el permiso para copiar, distribuir y / o modificar este documento bajo los términos de la Licencia de *Free Documentation License* de GNU, Versión 1.3 o cualquier versión posterior publicada por la Free Software Foundation.

*Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation.*



Figura 7.1. GFDL

## 8. Ley de protección de datos

La Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales (LOPDGDD) [10], desarrolla el REGLAMENTO (UE) 2016/679 DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 27 de abril de 2016 relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos.

Todos los formularios que aparezcan en la web de Serptracker donde se recoja información personal de sus usuarios deberán estar adaptados de manera que cumplan con el reglamento vigente, como por ejemplo el formulario de registro o el de contacto.

Tanto la política de privacidad como el aviso legal de Serptracker están a disposición de cualquier usuario que visite la web, pudiéndose informar estando o no registrado en la herramienta.

- **Política de Privacidad:** especifica las intenciones, y el fin de los datos recogidos en el sitio web Serptracker.
- **Aviso Legal:** se trata de una parte de la web de Serptracker donde se especifican los derechos del usuario y del autor de Serptracker.

## 9. Conclusión

El desarrollo de la aplicación Serptracker ha supuesto todo un desafío, ya que implicaba la utilización de lenguajes poco utilizados durante la carrera, como por ejemplo Javascript. La utilización de NodeJS ha sido muy satisfactorio, puesto que la concurrencia de acciones que ofrece es algo que hace las aplicaciones muy rápidas. Actualmente está muy extendido y existen numerosas ofertas tanto de programador *front-end* como *back-end* en este *framework*.

Por otro lado, como lenguaje de *scripting*, Ruby ha resultado ser muy útil junto con la librería de *web scraping* llamada *Nokogiri*, dado que los datos y los *proxys* son obtenidos con *scripts* desarrollados en este lenguaje.

Si bien es cierto que estos lenguajes de programación no se han utilizado apenas en la carrera, durante toda ella se nos ha enseñado a tener una base y una visión de las cosas que nos permite adaptarnos a todas las situaciones y aprender rápidamente cualquier cosa nueva.

En cuanto a la utilidad, se pensó en esta idea porque ya tenía una base de SEO aprendida de manera autodidacta durante años y tenía la necesidad de utilizar una herramienta como esta. La idea era combinar lo aprendido en Ingeniería Informática para utilizar nuevos lenguajes con el SEO y se ha conseguido.

Este Trabajo de Fin de Grado de Ingeniería Informática ha utilizado terminologías aprendidas en asignaturas como base de datos, seguridad informática, ingeniería del software, interacción persona ordenador o diseño y administración de redes. Ha sido muy fructífero a nivel de conocimientos y cercano al mercado laboral, puesto que las tecnologías utilizadas están demandadas hoy en día.

## 10. Referencias bibliográficas

- [1] Chart.js | Open source HTML5 Charts for your website. (2017). [Internet]. [actualizado 19 may 2017. 22 sep 2019]. Disponible en: <https://www.chartjs.org/>
- [2] Class: Net::HTTP (Ruby 2.6.5). (2015). [Internet]. [actualizado 11 mar 2015. 12 sep 2019]. Disponible en: <https://ruby-doc.org/stdlib-2.6.5/libdoc/net/http/rdoc/Net/HTTP.html>
- [3] Dalessio, M. (2016). Parsing an HTML/XML document - Nokogiri. [Internet]. [actualizado 05 mar 2016. 11 jun 2019]. Disponible en: [https://nokogiri.org/tutorials/parsing\\_an\\_html\\_xml\\_document.html](https://nokogiri.org/tutorials/parsing_an_html_xml_document.html)
- [4] Express 4.x - Referencia de API. (2019). [Internet]. [actualizado 05 mar 2019. 24 jul 2019]. Disponible en: <https://expressjs.com/es/4x/api.html>
- [5] Funcional, A., desempeño, E., eCOMITIA 2.0, n., Olucaro Dashboard 1.0, n., Prometheus IDS Core 1.0, n., Cibersad y SIXA de Enxenio, n., & ProEducative 3.0, p. (2019). ISO 25010. [Internet]. [actualizado 14 may 2019. 28 nov 2019]. Disponible en: <https://iso25000.com/index.php/normas-iso-25000/iso-25010>
- [6] Lenguaje de Programación: Concepto, Tipos y Ejemplos. (2019). [Internet]. [actualizado 22 jul 2019. 01 sep 2019]. Disponible en: <https://concepto.de/lenguaje-de-programacion/>
- [7] Mark Otto, a. (2019). Introduction. [Internet]. [actualizado 18 mar 2019. 19 oct 2019]. Disponible en: <https://getbootstrap.com/docs/4.1/getting-started/introduction/>
- [8] SSH Tunnel - Local and Remote Port Forwarding Explained With Examples - sensible.io Blog. (2019). [Internet]. [actualizado 01 dic 2018. 11 jun 2019]. Disponible en: <https://blog.sensible.io/2014/05/17/ssh-tunnel-local-and-remote-port-forwarding-explained-with-examples.html>
- [9] Threads in NodeJs — Performance Optimization. (2019). [Internet]. [actualizado 05 jun 2019. 08 sep 2019]. Disponible en: <https://medium.com/tech-tajawal/threading-in-nodejs-5d966a3b9858>
- [10] Traffic, P. (2016). Proxy for Ruby HTTP traffic. [Internet]. [actualizado 11 mar 2016. 14 jul 2019]. Disponible en: <https://stackoverflow.com/questions/22236591/proxy-for-ruby-http-traffic>
- [11] User Agent Identifier | Heroku Dev Center. (2018). [actualizado 29 abr 2018. 22 ago 2019]. Disponible en: <https://devcenter.heroku.com/articles/useragentidentifier#using-with-ruby>

- [12] StarUML Documentación. (2019). [actualizado 10 jul 2019. 18 oct 2019]. Disponible en: <https://docs.staruml.io/>

## 11. Anexos

### 11.1. MANUAL DE USUARIO

El siguiente manual explica el funcionamiento de Serptracker, desde cómo crear una cuenta de usuario hasta la creación de un proyecto y seguimiento de los resultados.

El primer paso para acceder a la herramienta Serptracker es dirigirse a su página web:  
<https://serptracker.es/>

#### 11.1.1. Registro

Si el usuario no posee una cuenta, deberá registrarse. Deberá dirigirse al apartado “**registrar cuenta**” que aparece debajo del botón de *login*.

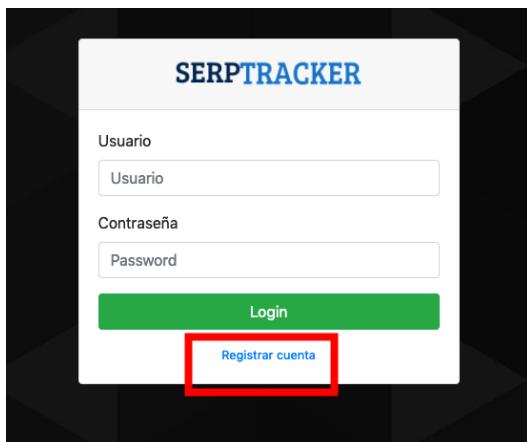


Figura 11.1.1.1. Apartado donde dirigirse para registrar una nueva cuenta.

Una vez se ha accedido al apartado de registrar cuenta, deberán rellenarse el correo electrónico, el nombre de usuario y la contraseña.

A screenshot of the "Registrar nueva cuenta" (Register new account) form. The form has a light blue header bar with the title and a "Volver al login" (Return to login) link. Below the header are three input fields: "Email" containing "666003@unizar.es", "Nombre de usuario" (User name) containing "jorge", and "Contraseña" (Password) containing ".....". At the bottom of the form is a blue "Registrar cuenta" (Register account) button.

Figura 11.1.1.2. Formulario de registro de una cuenta de usuario.

Si tras pulsar el botón de “Registrar cuenta” aparece un mensaje indicando que la cuenta se ha creado, ya podremos dirigirnos al apartado de “login”. En caso contrario, el error vendrá de que hemos elegido un nombre de usuario ya existente.

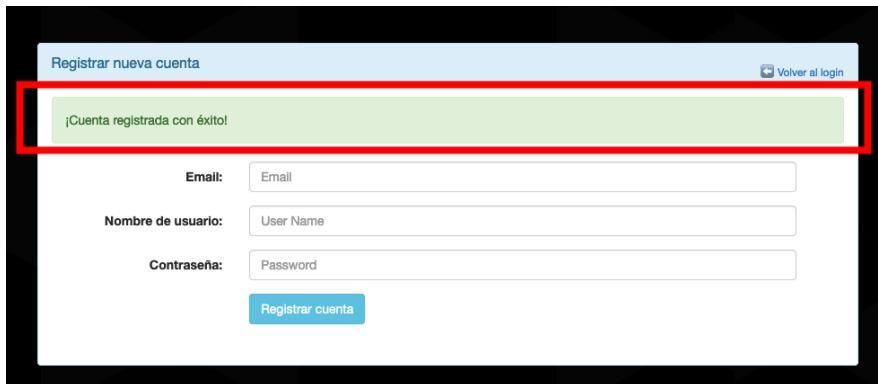


Figura 10.1.1.3. Mensaje indicativo de que la cuenta ha sido creada.

### 11.1.2. Login

El “login” corresponde con el inicio de sesión en el sistema. El usuario deberá introducir el usuario y la contraseña con los que se registró.

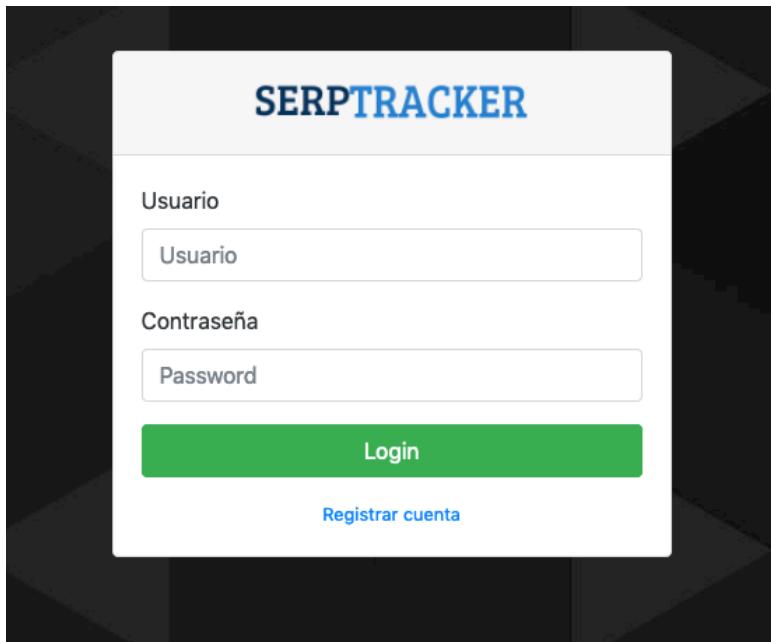


Figura 11.1.2.1. Login de usuario.

Si los datos de acceso son incorrectos aparecerá un error y tendrá que volver a intentarlo.



Figura 11.1.2.2. Mensaje de error por credenciales incorrectas.

### 11.1.3. Añadir nuevo proyecto

Para añadir un nuevo proyecto, el usuario debe dirigirse al apartado de “Proyectos→Nuevo proyecto”

Una vez allí, tendrá que llenar los siguientes apartados:

- **Nombre del proyecto.**
- **URL:** dirección de la página web de la que se realizará el seguimiento.
- **Palabras clave:** términos para los cuales se realizará el seguimiento en Google. Deben colocarse uno por línea.

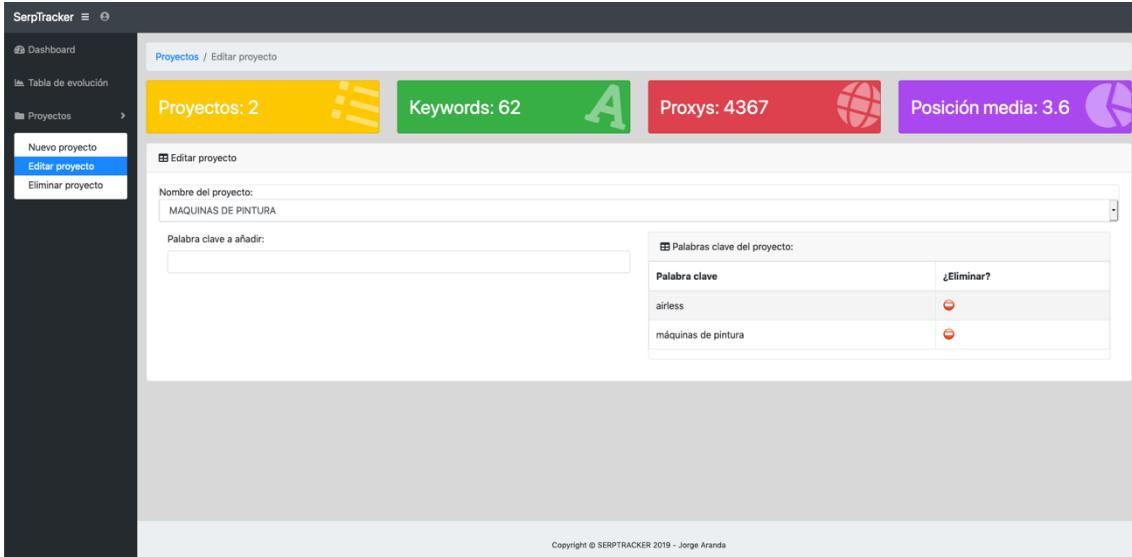
A screenshot of the 'Nuevo proyecto' (New project) creation page in the SERPTRACKER dashboard. The left sidebar shows navigation options like 'Dashboard', 'Tabla de evolución', 'Proyectos', 'Nuevo proyecto' (which is highlighted in blue), 'Editar proyecto', and 'Eliminar proyecto'. The main area has a header 'Proyectos / Nuevo proyecto'. It displays summary statistics: 'Proyectos: 2' (yellow), 'Keywords: 62' (green), 'Proxys: 4367' (red), and 'Posición media: 3.6' (purple). Below these are input fields for 'Nombre del proyecto:' (Project name), 'URL:', and 'Palabras clave:' (Keywords). The 'Palabras clave:' field contains 'Palabra clave 1' and 'Palabra clave 2'. A blue 'Crear proyecto' (Create project) button is at the bottom. The footer credits 'Copyright © SERPTRACKER 2019 - Jorge Aranda'.

Figura 11.1.3.1. Página de creación de un nuevo proyecto.

#### 11.1.4. Editar proyecto existente

Para editar un proyecto existente se debe acceder a “Proyectos→Editar proyecto”. Una vez ahí, se debe seleccionar el proyecto a editar, pudiendo así eliminar o añadir palabras clave a dicho proyecto.

Para eliminar una palabra clave es necesario pulsar sobre el icono “” situado a la derecha de la palabra clave.



Palabras clave del proyecto:	
Palabra clave	¿Eliminar?
airless	
máquinas de pintura	

Figura 11.1.4.1. Página para editar proyectos y añadir o eliminar palabras clave.

#### 11.1.5. Eliminar un proyecto

Para eliminar un proyecto debe accederse al apartado de la web “Proyecto→Eliminar proyecto”.

Seleccionando el proyecto a eliminar y pulsando el botón de “eliminar” se elimina un proyecto y todos sus datos registrados.

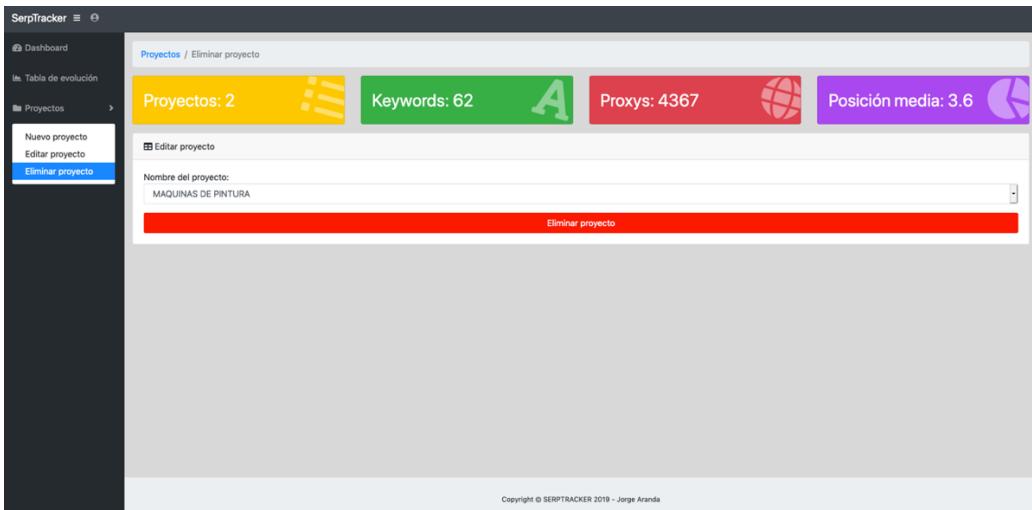


Figura 11.1.5.1. Página para eliminar proyectos.

### 11.1.6. Tipos de gráficas

Serptracker posee dos formas de visualizar la evolución de posiciones, desde el “Dashboard” o panel de control principal y la opción de ver la “tabla de evolución”:

- **Dashboard:** es la página de inicio de la herramienta y muestra en forma de gráfica [1] lineal todos los datos recogidos por la aplicación para el proyecto seleccionado.

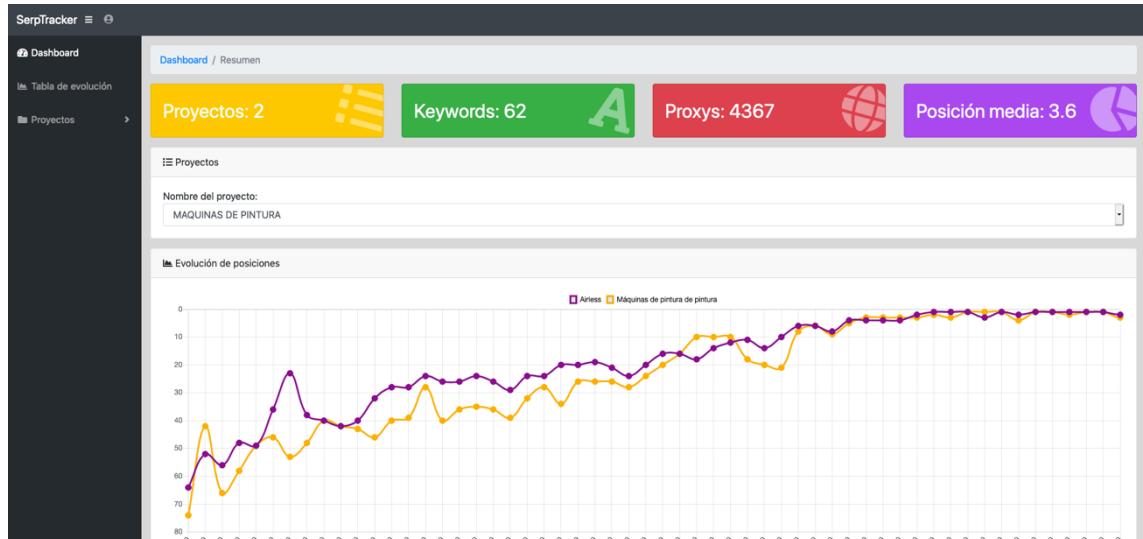


Figura 11.1.6.1. Dashboard/página de inicio de la herramienta.

- **Evolución de proyectos:** esta opción permite visualizar en forma de tabla la evolución de posiciones de las palabras clave del proyecto seleccionado indicando en las columnas la posición actual, la del día anterior, la de hace una semana y la de hace un mes. También indica entre paréntesis el número de posiciones ganas o perdidas respecto a dicho día.

The screenshot shows the "Tabla de evolución" section of the Serptracker interface with the following data:

Palabra clave	URL indexada	Posición	Día	Semana	Mes
airless	<a href="https://www.maquinasddepintura.com/airless/">https://www.maquinasddepintura.com/airless/</a>	1	1 (0)	4 (-3)	26 (-25)
máquinas de pintura	<a href="https://www.maquinasddepintura.com/">https://www.maquinasddepintura.com/</a>	2	2 (0)	6 (-4)	36 (-34)

Figura 11.1.6.2. Tabla de evolución de Serptracker.