# API para la generación de imágenes de tipo “shoppable image”

Ana Aguilera Ramos

**Resumen–** El e-commerce es una de las tendencias más relevantes para el sector del comercio y lo lleva siendo desde hace más de 10 años gracias a la existencia de Internet y las empresas que dieron inicio a esta nueva forma de hacer negocios. Las redes sociales y las páginas web se llenan de imágenes de productos que la gente quiere obtener, el escaparate de pequeños y grandes comercios se extiende de manera significativa. Una nueva tendencia surge, las shoppable images con las que se pretende integrar una nueva manera de dar a conocer productos a través de imágenes de publicaciones acompañadas de tags informativos. Con esto se pretende generar una página web donde los usuarios tengan la libertad para generar dichas imágenes, permitiendo su personalización y además el uso de estas en su página web, perfil, etc de manera fácil a través de la implementación de una API que permita gestionar el acceso de los dominios que soliciten este servicio.

**Palabras clave–** Comercio electrónico, shoppable-images, API KEY, Scrum , Kanban, SaaS, Software como servicio

**Abstract–** E-commerce is one of the most relevant trends for the commerce sector and it has been so for more than 10 years thanks to the existence of the Internet and the companies that started this new way of doing business. Social networks and web pages are filled with images of products that people want to obtain, the showcase of small and large businesses is expanding significantly. A new trend arises, the shoppable images with which it is intended to integrate a new way of publicizing products through images of publications accompanied by informative tags. With this, it is intended to generate a web page where users have the possibility to generate these images, allowing their personalization and also the use of these in their web page, profile, etc. in an easy way through the implementation of an API that allows managing the access of the domains that request this service.

**Keywords–** e-commerce, shoppable-images, API KEY, Scrum , Kanban, SaaS, Software as a service

## 1 Introducción

El contenido de este artículo pretende documentar el desarrollo de una página web que permita la generación de shoppable-images para que posteriormente los usuarios puedan hacer uso de estas en su propia página web a través de la conexión a una API que limitará su servicio a ciertos dominios autenticando su identidad.

Además, se pretende mostrar la adaptación del uso de la metodología ágil de Scrum y los tableros Kanban para un trabajo individual donde se deben generar incrementos y documentación complementaria.

A continuación, se procede a indicar detalladamente el proceso de desarrollo de esta página web, dar contexto a la importancia de este nuevo concepto de shoppable-image en el Internet de nuestra generación, especificar la manera de trabajar para lograr buenos resultados y realizar una valoración adecuada al incremento generado y su funcionamiento final además de incluir posibles mejoras y ampliaciones de esta a futuro.

• E-mail de contacte: ana.aguilera.99@gmail.com
• Mención realitzada: Tecnologies de la Informació
• Treball tutoritzat per: Marc Ortega Gil (Departamento de Ingeniería de la Información y las Comunicaciones)
• Curs 2022/23

## 1.1. Estado del arte

Las tendencias actuales del sector del comercio se dirigen a la adopción e implantación de las nuevas herramientas que permiten el e-commerce debido a su alto impacto beneficioso para acaparar un mercado potencial como es internet y las compras online.
La llegada del e-commerce tiene sus primeros inicios con la llegada del Electronic Data Interchange en los años 60 donde existían ya los catálogos electrónicos, pero acaba de formarse con la llegada de Internet y la liberación de restricciones de este para uso con fines comerciales entre 1989 y 1991.
Empresas como Amazon y eBay tienen sus inicios durante los años siguientes. El comercio online sigue creciendo y evoluciona con las tecnologías. Esto desencadena en las tendencias actuales de e-commerce donde comenzamos a tener en cuenta los dispositivos conectados a Internet, como smartphones entre muchos otros más.
Este escenario se ve favorecido estos últimos 10 años debido a la gran cantidad de objetos que nos rodean y se conectan a Internet.
Además, existe una necesidad de los usuarios en ganar eficiencia en las compras mientras satisfacen sus necesidades comerciales.

Las publicaciones en redes sociales han comenzado a tener un gran valor para el comercio de productos o artículos. Las redes se llenan de imágenes donde las personas muestran productos como ropa, productos de belleza, utensilios de cocina, etc. Ya no son simplemente las marcas las que se encargan de generar las tendencias en el gran escaparate de Internet y redes.
Las personas públicas y no tan públicas tienen la necesidad de compartir sus compras, adquisiciones o recomendaciones con la comunidad. Esto también hace que tiendas online intenten adoptar nuevas tendencias como el uso de las shoppable-images.

Estas shoppable-images son una nueva herramienta para mostrar estos productos de manera limpia y clara haciendo accesible de forma inmediata con un pop-up ciertas características como el nombre del producto, marca y enlace de compra entre otros más simplemente pasando el cursor sobre los productos que aparecen en las imágenes.
La idea principal de las shoppable-images es hacer de una imagen una herramienta que permita ampliar el escaparate de los productos de las empresas y que estos puedan llegar a más personas.
La restricción de publicitar los productos solamente desde la web donde se tienen a la venta desaparece para dar paso a un escenario donde cualquier imagen en cualquier sitio web en Internet puede tener la posibilidad de llevarte a esta tienda para comprarlo si por un casual encuentras una publicación donde estos aparecen.

## 1.2. Objetivos

El interés en la herramienta de las shoppable-images parece ir en aumento, ya que empresas como Amazon o Google están comenzando a adoptar estas en ciertas regiones y en ciertas secciones diferenciadas de sus plataformas.
La mejora en las ventas de las tiendas online a través de la generación de estas imágenes que se pueden integrar en cualquier sitio web es suficientemente significativa como para generar interés en los usuarios, haciéndoles plantearse la posible adopción y uso de estas.

Con la implementación de la web de generación de estas shoppable-images se pretende facilitar el trabajo a aquellos que quieran comenzar a integrar esta herramienta en sus páginas web.
Además de la generación de las shoppable-images se pretende añadir opciones de personalización para que estos usuarios puedan generar sus imágenes y además darles un aspecto distintivo en relación a la estética de su propia página web.

La web por implementar permitirá a los usuarios registrarse para poder obtener acceso al uso de esta herramienta de creación/edición de proyectos.
En ella se podrán generar proyectos, donde estos serán una shoppable-image.
Los proyectos serán diferenciados por un identificador único que será una API KEY y podrán ser guardados para su posterior edición.
Los proyectos permitirán al usuario subir una imagen de su gusto y trabajar sobre ella añadiendo tags para marcar los productos que esta contiene y añadir información sobre estos (Título, Descripción corta( Precio, Marca...), URL del producto).
Los tags, la tipografía, colores entre otras opciones podrán ser configurados también en este apartado al gusto de los usuarios.
Finalmente, un fragmento de código podrá ser generado, de manera que el usuario pueda visualizarlo y copiarlo desde el editor de proyectos para su posterior uso junto a la API KEY.
Este proyecto debe generar una visualización de las shoppable-images diferenciada para ordenador y smartphone, siendo esto transparente para el usuario, ya que será detectado pasivamente, en ambas esta deberá ser responsive, es decir, que se adapte al iframe que lo contiene si este cambia de tamaño.

En conclusión, queremos implementar una web para la generación de estas imágenes de manera que su integración sea fácil, clara y personalizable.

Para la planificación inicial se añadieron los siguientes objetivos generales:

- Familiarización y análisis para la planificación del desarrollo de la web a través de la lectura de documentación.
- Análisis de las necesidades del BackEnd (definición de las funcionalidades a implementar).
- Análisis de las necesidades del FrontEnd (definición de las funcionalidades a implementar).
- Diseño de la página web y definición de sus funcionalidades.
- Implementación incremental de la parte BackEnd.

- Implementación incremental de la parte FrontEnd.
- Redacción de los informes y documentación.
- Análisis del trabajo realizado y reestructuración de las tareas si fuera necesario llevando un registro de estos cambios.
- Actualización del estado del tablero Kanban.
- Test del nuevo incremento de la página web y valoración del estado de progreso.

Todos estos objetivos generales fueron analizados y desencadenaron en la generación de una serie de tareas mucho más especificas. Todas ellas quedaron registradas en la herramienta de Trello.
Antes de continuar y ponernos con la metodología seleccionada queríamos destacar que esta API se plantea como un SaaS, Software como servicio página web.

### 1.3. SaaS

Software as a service o Software como servicio es un modelo de distribución de software que pretende ofrecer una funcionalidad o software concreto a usuarios externos asumiendo íntegramente el cómputo siendo el host donde esto se ejecuta y proporcionando acceso de esta a usuarios a través de internet.
De esta manera los usuarios pueden obtener acceso a esta funcionalidad de manera externa sin tener que programarla y ejecutarla ellos mismos ahorrándose cargas innecesarias.
Este es un modelo de servicio bastante generalizado el cual normalmente es de pago a través de suscripciones mensuales.
En nuestro caso ofrecemos el servicio para la creación de las shoppable-images, además de su personalización y visualización.
Este servicio sería compatible con este modelo de negocio y podría llegar a tener futuro si se realizará su versión comercial extendiendo sus funcionalidades.

Una vez planteada esta introducción procedemos a especificar la metodología seguida que permitiría ordenar y desarrollar los objetivos durante el desarrollo real para conseguir llevar a cabo esta idea.

## 2 METODOLOGÍA

Para el desarrollo de este proyecto era necesario establecer una manera de trabajar adecuada, de manera que nos permitiera marcar objetivos reales que pudieran generar un incremento del código marcando periodos diferenciados teniendo en cuenta las fechas de entrega y además admitiera replanificar en cierto grado la planificación si era necesario en algún momento.
Durante nuestro paso por el grado hemos trabajado muchas de estas metodologías ágiles cosa que ha permitido que tengamos un pensamiento crítico a la hora de escoger una de ellas para trabajar.
En nuestro caso decidimos utilizar la metodología ágil de Scrum junto a tableros Kanban.

La metodología Scrum nos permitiría marcar los periodos de tiempo diferenciados (Sprints), un listado de objetivos generales inicial y a su vez objetivos específicos para los Sprints que marcarían los incrementos deseados al final de estos y con ello generar una valoración del progreso global del proyecto.
Además el uso de los tableros Kanban permitirían tener un soporte visual donde poder ver las tarjetas correspondientes a los temas y las subtareas más detalladas que corresponderían a las tareas específicas a realizar.

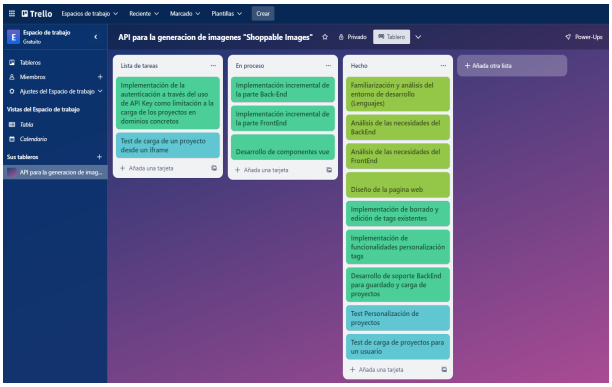


Fig. 1: Pagina web de Trello (Kanban)

Con estas dos cosas el control del flujo de desarrollo del proyecto quedaría cubierto.
Una de las cosas más importantes a realizar era la planificación inicial donde se marcaron las tareas más importantes con las que dar inicio al desarrollo además de la elección de los lenguajes de programación con los que se realizaría la implementación de este proyecto.
Esta elección era un tema crítico, ya que una vez iniciado el desarrollo no podría ser modificado, ya que tendría un impacto negativo provocando un retraso en el desarrollo.

A continuación, podemos ver la tabla generada para marcar los tiempos de estos Sprints:

TABLA 1: TABLA DE SPRINTS

| Sprint | Periodo |
|---|---|
| Sprint-1 | 13/03 a 26/03 |
| Sprint-2 | 27/03 a 09/04 |
| Sprint-3 | 10/04 a 23/04 |
| Sprint-4 | 24/04 a 07/05 |
| Sprint-5 | 08/05 a 21/05 |
| Sprint-6 | 22/05 a 04/06 |
| Sprint-7 | 05/06 a 18/06 |
| Sprint-8 | 19/06 a 02/07 |
| Sprint-9 | 03/07 a 16/07 |

A continuación, damos paso a la explicación del desarrollo y el trabajo realizado.

# 3 DESARROLLO

Esta sección pretende dejar reflejadas las diferentes fases de desarrollo del proyecto, además de aclarar ciertos conceptos si es necesario.
Para este desarrollo se escogieron unos lenguajes de programación específicos:

- HTML: Lenguaje estándar W3C para la creación de páginas web.
- CSS: Lenguaje “style sheet” para la modificación del diseño de documentos HTML o XML.
- JavaScript: Lenguaje de programación interpretado utilizado para crear páginas webs interactivas entre otras cosas como aplicaciones backend.

Estos son los 3 lenguajes core en la creación de páginas webs existentes en internet, además de ser lenguajes fáciles de aprender para principiantes. Existen otros lenguajes como C, Python, Go entre otros, pero algunos tienen tiempos de compilación altos, otros son poco utilizados o complejos de aprender y consideramos que utilizando estas 3 opciones indicadas anteriormente era suficiente para lo que se pretendía implementar.

## 3.1. Diseño

Para dar comienzo al desarrollo del proyecto una vez seleccionados los lenguajes de programación se dio paso a una primera fase donde realizaríamos el diseño funcional y un esbozo del diseño visual de la página.
Después de un previo análisis de necesidades acorde a las funcionalidades deseadas para la página web y la API se comenzó a generar el esqueleto inicial del código configurando el entorno e instalando los módulos necesarios.

El diseño general de las comunicaciones y los módulos necesarios fue el siguiente:

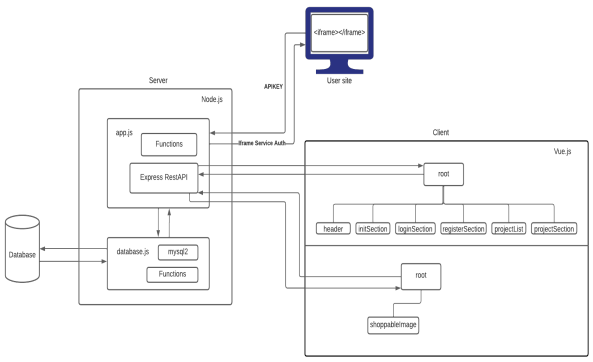


Fig. 2: Esquema de comunicaciones

En este esquema podemos observar dos módulos diferenciados:

- El módulo de Servidor: contendría el BackEnd implementado con Node.js que comprende la comunicación entre el cliente, la base de datos y las comunicaciones de la API.
- El módulo de Cliente: contendría el FrontEnd implementado con Vue.js, CSS y HTML que comprende la parte visual e interactiva de la página web a desarrollar y la visualización del contenido para el formato de acceso externo.

Las comunicaciones se realizaron a través del uso del package Express de node.js que se utiliza generalmente para generar aplicaciones de BackEnd.
Se encarga de los detalles del Backend, como las sesiones, la gestión de errores y el enrutamiento. Proporciona un conjunto de instrumentos para aplicaciones web, solicitudes HTTP y respuestas entre otras cosas más.

Como se puede observar en la Figura 2 era necesaria la existencia de una base de datos donde se guardaran las cuentas y proyectos de los usuarios que usen la página.
Para ello decidimos instalar XAMPP Control Panel v3.3.0 para la gestión de la base de datos MySQL que requeríamos crear configurando el host localmente.
Además se utilizó la herramienta de HeidiSQL v12.1.0.6537 para todo lo relacionado con el desarrollo y administración de la base de datos.

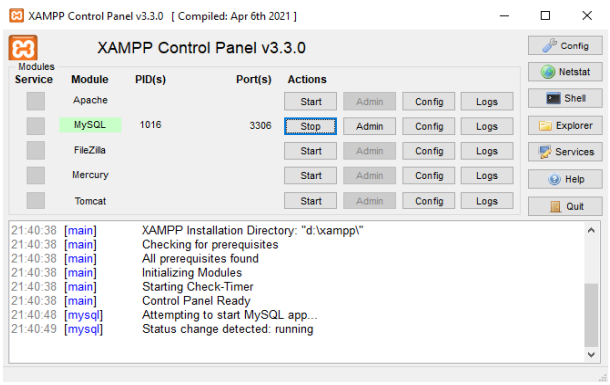


Fig. 3: Herramienta XAMPP

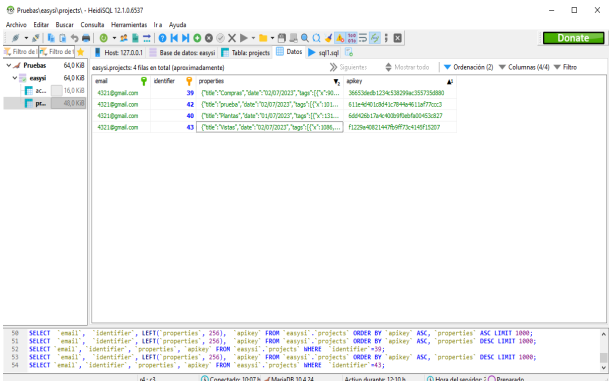


Fig. 4: Herramienta HeidiSQL

En esta base de datos después de analizar las necesidades futuras de la aplicación decidimos crear dos tablas, accounts y projects.

La tabla de accounts guarda el email del usuario, la contraseña cifrada y el dominio a verificar, por otro lado la tabla de projects guarda el email, el identificador de proyecto, las propiedades guardadas al realizar la personalización de este y la API KEY correspondiente a dicho

proyecto.

Una vez generado el soporte para la base de datos se comenzó a generar la parte del servidor correspondiente a node.js que se encargaría de procesar y gestionar los datos entre la SQL y la parte visual del Cliente.

Se instalaron los siguientes paquetes necesarios (dependencias):

- bcrypt: v5.1.0
- express: v4.18.2
- express-session: v1.17.3
- generate-api-key: v1.0.2
- mysql2: v3.2.3
- sanitize-html: v2.10.0
- ...

Aquí mencionamos algunos, pero muchos paquetes fueron instalados y estos están especificados en nuestro archivo del código package-lock.json.

Generamos el archivo app.js que engloba toda la parte del servidor, creamos una app con Express y la pusimos a escuchar de manera local al puerto 3000. En este punto aún no se había contemplado la posibilidad de generar una segunda aplicación que se encargara exclusivamente del servicio externo de visualización, pero más tarde se terminaría implementando (appAPI.js).

Una vez realizada esta parte la conectamos con el archivo central index.html que conectaría con el script de la parte de vue.js donde nosotros nos encargaríamos de implementar los componentes necesarios y todo el funcionamiento de las vistas para que fuera fiel a la idea planteada inicialmente.

El desarrollo de la implementación debía realizarse de manera que se generaran tanto la parte del FrontEnd como del BackEnd para hacer posible el test de las funcionalidades previstas para el final de cada Sprint, con lo cual se decidió realizar esbozos simples de lo que serian las diferentes vistas que tendría la página web para poder hacernos una idea de su aspecto y además generar las funcionalidades del BackEnd que complementarían su funcionamiento y aportarían los datos necesarios para su visualización.

Como no era necesario perder tiempo en generar algo muy detallado decidimos crear con ayuda de Photoshop CS6 un esbozo minimalista que englobara todas las vistas de la página web:

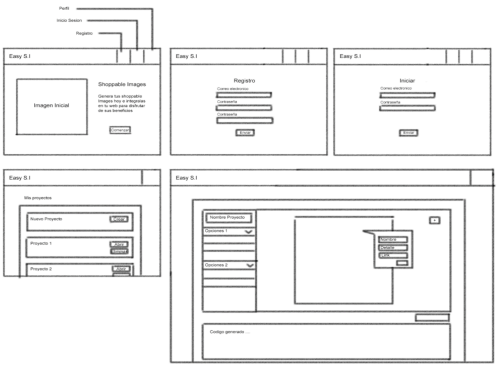

Fig. 5: Boceto simple de vistas

Con esto generamos una idea global de lo que sería la página web sin entrar demasiado en profundidad para poder invertir el tiempo en la posterior implementación.

## 3.2. Implementación

Una vez generadas las especificaciones y hecho el esbozo inicial de las vistas procedimos a dar comienzo a la implementación de los objetivos propuestos comenzando por la parte del FrontEnd generando datos de ejemplo iniciales para probar y más tarde generamos la parte del BackEnd que generaría y cargaría los datos conectándose a la SQL.
En un primer directorio llamado public englobaríamos css, vue.js e imágenes por separado.
Para mantener la distribución generaríamos también un directorio a parte llamado model donde tendríamos el archivo database.js para mantener diferenciado y localizado todo lo relacionado con querys de la base de datos.
Todo este código fue subido a la plataforma de GitHub en un repositorio privado para asegurarnos de tener una copia de seguridad en caso de ocurrir algún inconveniente y además generar versiones diferenciadas, de manera que pudiéramos recuperar el estado de los archivos si requeríamos volver a algún punto utilizando GitBash en el directorio correspondiente al repositorio.

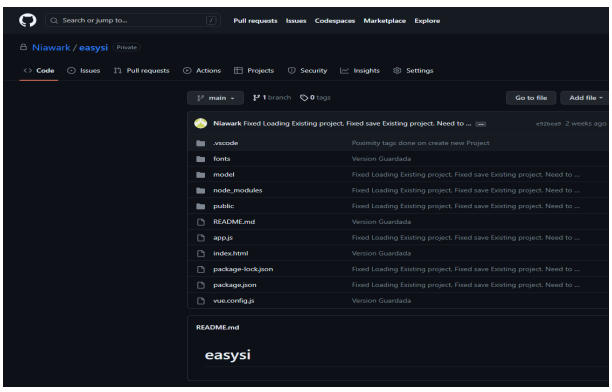


Fig. 6: Repositorio GitHub

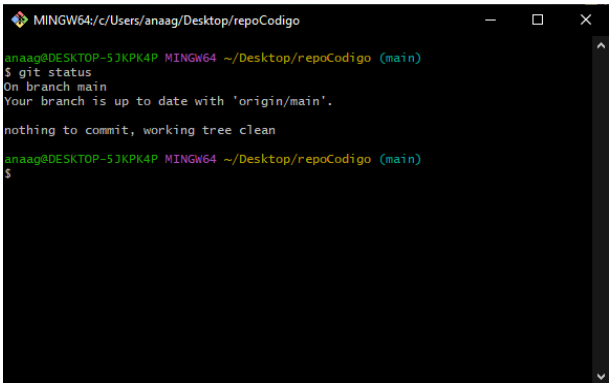


Fig. 7: Consola de GitBash

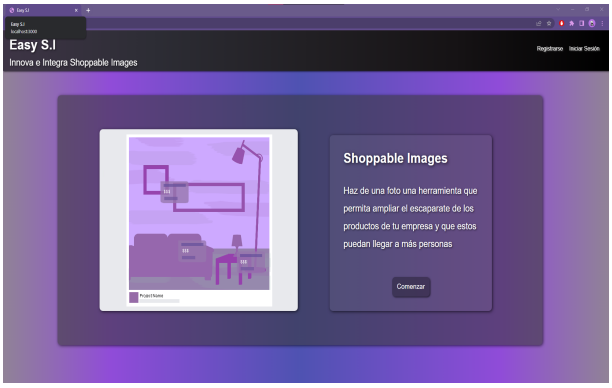


Fig. 8: Pantalla de inicio

La aplicación de git bash también nos permitió ejecutar los scripts a través del comando:

- node scriptExample.js

De esta manera podíamos tener los scripts que necesitáramos ejecutando en consolas separadas donde ver los errores si es que surgía algún caso.
Después de toda esta parte se comenzó a pensar más en profundidad en la estética de la página.
Se decidió utilizar tonos violetas para la página web, además de un aspecto limpio, simple y minimalista.

Los componentes a implementar eran los siguientes:

- headerComponent
- projectlistSection
- initialSection
- registerSection
- loginSection
- tagComponent
- projectSection
- alertPopUp

Cabe destacar que se decidió separar la implementación en dos aplicaciones diferentes una vez llegado el momento, la primera sería la aplicación de la página web y la segunda sería la aplicación de la carga de las shoppable Images de manera que los accesos de la página quedaran separados de esta parte del servicio asignándoles puertos diferentes 3000 y 3001.
La implementación comenzó generando la pantalla de inicio de la aplicación junto a la barra fija del header que se mantendría en todas las vistas.

Esta parte simplemente debía permitir al usuario conocer el concepto de shoppable image y conectar con las opciones de registro e inicio de sesión que posteriormente dejarían acceder al usuario a la sección del listado de proyectos donde se listarían los ya creados junto a la opción de creación de un proyecto nuevo:

Seguidamente, se implementó la parte del registro y el inicio de sesión, en primer lugar la parte visual junto al estilo deseado y poco después la parte funcional que se encargaría de guardar los datos de la cuenta del usuario de manera segura utilizando el cifrado de bcrypt.
Los datos introducidos pasan por un sanitizeHtml que se asegura de evitar cualquier cadena de caracteres maliciosa que intente inyectar código.

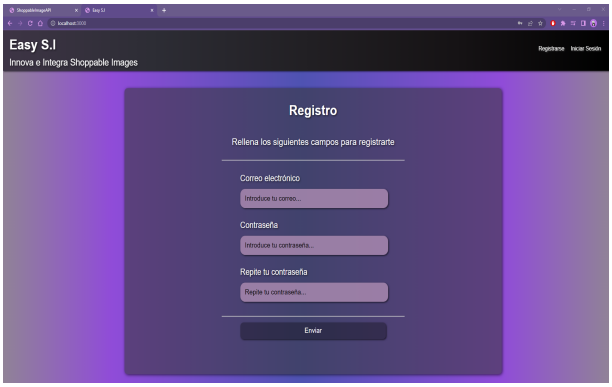


Fig. 9: Registro

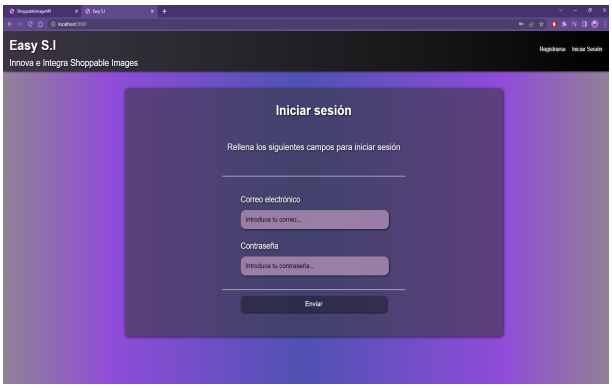


Fig. 10: Inicio de sesión

Por lo que se puede observar ambos son muy parecidos, al final los dos son formularios que nos permiten generar y verificar la identidad de los nuevos usuarios entrantes.
Una vez conseguido el registro y el inicio de sesión se generaron datos de ejemplo de proyectos para iniciar con la sección del listado y poder implementar los botones de creación, carga y borrado de proyectos.

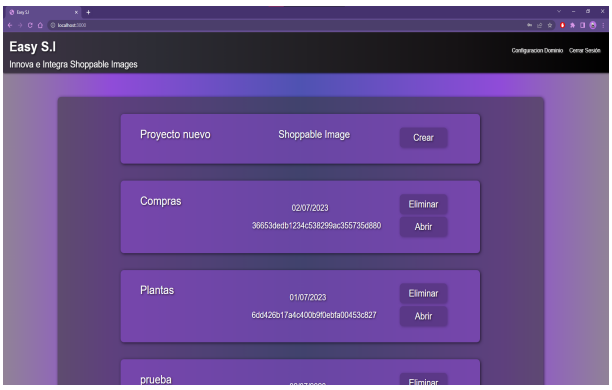


Fig. 11: Listado de proyectos

La parte de estos componentes fue la más sencilla de implementar, a partir de este punto incrementó la dificultad. Los componentes de projectSection y tagComponent eran necesarios para la parte de creación de proyectos, donde los usuarios serían capaces de generar la personalización de sus imágenes y de los tags que escogieran añadir.
Para mostrar un ejemplo de funcionamiento se decidió simular la manera en que se mostrarían los tags en el mismo editor. Estos debían cargar dependiendo de la proximidad del cursor del usuario sobre la imagen, revisando las coordenadas constantemente mientras este se encuentre sobre la imagen.

El componente projectSection se dividió en 3 secciones, visualizador de imagen, opciones de personalización y listado de tags activos.

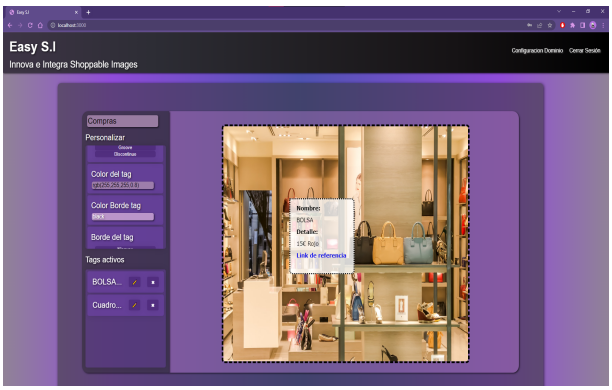


Fig. 12: Sección de edición A

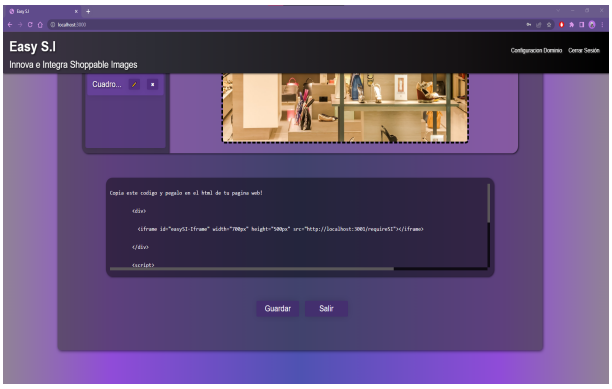


Fig. 13: Sección de edición B

Con estas tres secciones quedaron cubiertas las funcionalidades requeridas y permitió avanzar con el guardado y carga de proyectos al tener datos con los que trabajar.

Todo esto requirió de la implementación de la carga de los datos al servidor y posteriormente a la base de datos. También se descubrió la necesidad de identificar los proyectos indicando si estos eran de nueva creación o no en el componente de edición.
Una vez implementada esta parte se comenzó con la implementación de la generación de las API KEY correspondientes a cada proyecto, además del sistema de autenticación para su uso externo hosteado en un iframe.

La generación de esta APIKEY se decidió realizar con el paquete de generate-api-key: v1.0.2 en el momento donde se realizaba el primer guardado del proyecto.

Para esta parte se generó la segunda aplicación donde el único componente que pensamos implementar fue:

- shoppableImage

Este componente sería el encargado de hacer posible la conexión de una página externa y la página donde generamos el contenido.
Para ello se hizo uso del paquete cors, configurando el dominio permitido y un addEventListener en nuestro componente, mientras que en el código a cargar por parte del cliente se utilizó postMessage.

Toda esta simulación fue realizada de manera local.
Esto fue necesario para hacer posible el acceso de nuestro componente a las características del iframe que lo contendría, de manera que pudiéramos hacerlo responsive al tener el tamaño del iframe de antemano para trabajar con él.
Esto viene capado default por seguridad en los buscadores.

Para poder hacer tests de la funcionalidad de esta nueva aplicación se necesitó implementar una página externa (con un dominio local diferente) donde se cargaran los fragmentos de código generados.
Como se indicó, se pretendía realizar una versión visual para ordenador y otra para smartphone, ya que interactúan diferente.

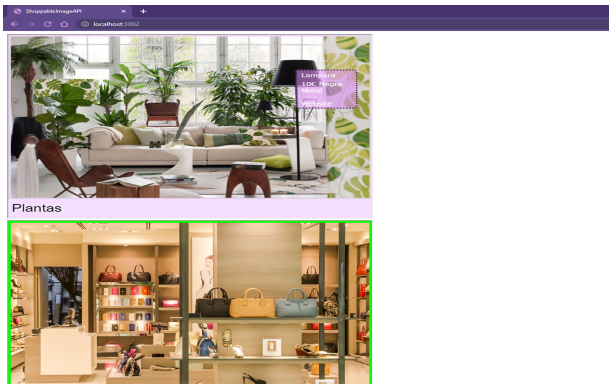


Fig. 14: Visualización ordenador

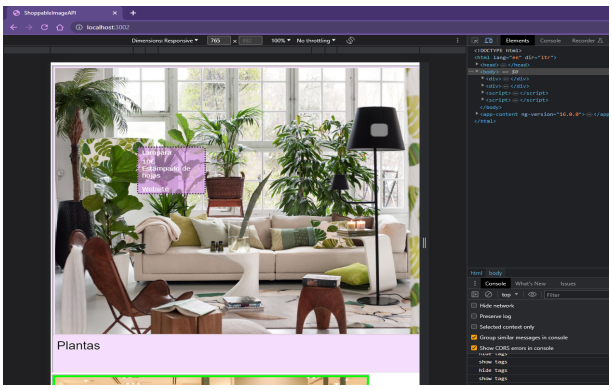


Fig. 15: Visualización smartphone

Como se puede observar, el comportamiento para la versión de ordenador es el mismo que en el simulador del editor de proyectos, mientras que para la versión de teléfono optamos por marcar las zonas que pueden contener un tag con un círculo translúcido gris para que los usuarios tengan una referencia visual de qué zonas pueden ser interactivas.

Esta funcionalidad fue probada a través del uso de las herramientas de desarrollador de Google Chrome que permite la simulación del dispositivo móvil.

## 4 Resultados

Los resultados obtenidos han sido en general buenos, aunque se podrían pulir ciertos aspectos de la implementación si nos centramos en algunos aspectos del css y la disposición de los elementos en el espacio que ha sido interesante de aprender usándolo en un proyecto más extenso de lo que nunca había realizado antes.
La página web es totalmente funcional, todas las acciones indicadas al inicio de este proyecto han sido satisfactoriamente conseguidas.
La primera parte, es decir, la página web, es totalmente funcional, exceptuando el apartado de configuración del dominio a utilizar por parte del usuario.
Esta limitación sí existe, pero no se ha realizado un componente de formulario para configurarlo al gusto del usuario.
Esto es debido a que la simulación ha sido ejecutada de manera local y no se ha priorizado la implementación de esta.
En todo caso a comentar que esta funcionalidad puede ser probada de forma manual si se cambia el dominio de la pagina externa y se actualiza en consecuencia en el lado del servidor.

En cuanto a la segunda parte, la visualización externa es correcta, sería mejorable a la hora de generar tags que se acercan a los bordes del iframe, ya que en ese caso estos quedan ocultos en parte, porque el contenido no sobresale de este como en la simulación del editor.
La parte visual ha quedado bastante estética, aunque hay alguna cosa que podría mejorar con animaciones que generen algo más de dinamismo en la página.
El seguimiento y la metodología de trabajo han sido de gran ayuda para mantener un ritmo constante en el desarrollo en solitario de este código.
Trabajar con estas en un caso real me ha demostrado la gran importancia que tienen estas para generar una la disciplina de consistencia y compromiso que proporciona grandes resultados de desarrollo.

## 5 Conclusiones

Finalizando este documento podemos decir que el desarrollo de las funcionalidades acordadas ha sido satisfactorio.
Ha sido un reto implementar todo de manera autónoma, pero ha sido muy bueno a nivel personal para mejorar en muchos aspectos y aprender como afrontar objetivos más grandes.
El servicio implementado tiene un gran potencial bajo mi punto de vista. Después de leer muchos artículos relacionados con las Shoppable Images y ver que empresas grandes están dándole una oportunidad podría llegar a implementarse a nivel comercial.
Sería bueno realizar un estudio de viabilidad para llevar este tipo de servicio al mercado real.

Para hostear de manera pública el servicio serían necesarios un dominio y el alquiler de un host remoto para ejecutarlo. Actualmente, existen muchas empresas que ofrecen este tipo de servicios con una suscripción mensual. Hay opciones que no son costosas para realizar esto, se pretende en un futuro mejorar e implementar esto si es posible, ya que ha sido un proyecto interesante de realizar y que ha hecho que despierte un interés en mí para desarrollar esta idea y ampliar mi conocimiento en el sector de los SaaS.
En conclusión, la implementación realizada tiene un potencial a futuro y sería interesante plantear en sólido la viabilidad de sacar al mercado este tipo de servicio.
He podido ver servicios relacionados con la edición de imágenes, pero no he encontrado ninguno parecido al concepto planteado en este proyecto.

## Agradecimientos

Agradecimiento a todas aquellas personas que han formado parte de mi formación y que me han apoyado siempre en el reto de superarme y aprender dentro del mundo de la Ingeniería.
Agradecer a mi tutor Marc Ortega por aconsejarme durante todo este tiempo en el seguimiento de mi trabajo.
Agradecer a mi familia que me apoya en todos mis retos y me mantiene enfocada en conseguir todo lo que me propongo.

En especial nombrar a mi Padre que habría querido verme cumpliendo mis objetivos en los que tanto esfuerzo he puesto siempre.

## Referencias

[1] El e-commerce: Sus orígenes, presente y tendencias futuras, Juan Luis González. [Online]. Available:

https://www.esic.edu/saladeprensa/prensa/noticia/el-e-commerce-sus-origenespresente-y-tendencias-futuras

[2] ¿Qué es el shoppable ads y por qué compite con Instagram? Vilma Núñez [Online]. Available: https://vilmanunez.com/que-es-el-shoppable-ads-y-por-que-compite-con-instagram/

[3] Google lanza los Shoppable Image Ads, Susana Galeano [Online]. Available: https://marketing4ecommerce.net/google-shoppable-image-ads/

[4] Introducing shoppable storefront images for amazon stores, Jennifer Sayroo [Online]. Available: https://marketing4ecommerce.net/google-shoppable-image-ads/

[5] Enable Interactive Product Tags, OSF Digital [Online]. Available: https://osf.digital/products/commerce-cloud/shoppable-images

[6] Shoppable images: What are they and how can they help increase sales?, Ricky [Online]. Available: https://www.idukki.io/blog/shoppable-images/

[7] Tell a story with Shoppable Photos, Amazon [Online]. Available: https://affiliate-program.amazon.com/resource-center/shoppable-photos-how-toand-best-practices

[8] The best programming languages you can use for web development, Lucero [Online]. Available: https://www.sitepoint.com/best-programming-language-for-web-development/html

[9] JavaScript, mdn web docs [Online]. Available: https://developer.mozilla.org/es/docs/Web/JavaScript

[10] npm Docs, npmjs [Online]. Available: https://docs.npmjs.com/

[11] Packages and Modules, npm Docs [Online]. Available: https://docs.npmjs.com/packages-and-modules

[12] CSS ,mdn web docs [Online]. Available: https://developer.mozilla.org/es/docs/Web/CSS

[13] HTML ,mdn web docs [Online]. Available: https://developer.mozilla.org/es/docs/Web/HTML

[14] Progressive JavaScript Framework, Vue.js [Online]. Available: https://vuejs.org/

[15] Trello, Atlassian [Online]. Available: https://trello.com/es