

Grado en Ingeniería

Telemática



Trabajo Fin de Grado

|  |
| --- |
| SellArt  La Web para Artistas |

**Autor:** Aarón Sanz Roca

**Tutor/a:** Javier Enrique Meseguer Anastasio

**Julio 2022**

****Declaración de autoría:****

Yo, Aarón Sanz Roca, declaro la autoría del Trabajo Fin de Grado titulado “SellArt: La Web para Artistas” y que el citado trabajo no infringe las leyes en vigor sobre la propiedad intelectual. El material no original que figura en este trabajo ha sido atribuido a sus legítimos autores.

Valencia, 1 de julio de 2022

Fdo: Aarón Sanz Roca

**Resumen:**

El avance de las tecnologías estos últimos años a dado fruto a que todas las facetas artísticas hayan sido obligadas a abrirse paso por la nube, y, por tanto, el mercado de estas ha tenido que seguir sus pasos. SellArt es la aplicación web que proporciona este camino, una tienda online creada para los artistas, con el objetivo de que los mismos puedan ganarse la vida pudiendo vender sus obras. Del mismo modo, por tanto, los amantes del arte podrán adquirir sus obras a través de la web, haciendo de la compra a través de internet, la opción más cómoda posible.

**Abstract:**

The advancement of technologies in recent years has resulted in all artistic facets have been forced to make their way through the cloud, and, therefore, the market of these has had to follow in their footsteps. SellArt is the web application that provides this way, an online store created for artists, with the aim that they can earn a living by being able to sell their works. In the same way, therefore, art lovers will be able to acquire their works through the web, making the purchase through the Internet, the most comfortable option possible.

Resum:

L'avanç de les tecnologies aquests darrers anys ha donat fruit a que totes les facetes artístiques hagin estat obligades a obrir-se pas pel núvol, i, per tant, el mercat ha hagut de seguir els seus passos. SellArt és l'aplicació web que proporciona aquest camí, una botiga en línia creada per als artistes, amb l'objectiu que aquests puguin guanyar-se la vida podent vendre les seves obres. De la mateixa manera, per tant, els amants de l'art podran adquirir les seves obres a través del web, fent de la compra a través d'Internet, l'opció més còmoda possible.

Agradecimientos:

Quiero agradecer a todos aquellos que me han hecho llegar hasta aquí, sin dejarme rendirme en ningún momento, siempre levantándome cuando yo estaba dispuesto a caer.

Agradecer principalmente a mis padres por preocuparse desde el primer día en mi salud y bienestar. A mis hermanos también, por enseñarme que siempre hay que seguir creciendo. Y finalmente, a mis amigos, por haberme sacado las sonrisas necesarias para haber hecho este camino más ameno, y especialmente, a aquellos que han estado, día tras día, obligándome a no dejarme esto para el día siguiente.

Gracias a todos por hacer esto posible <3

Índice general

Capítulo 1. Introducción 12

1.1 Introducción 12

1.2 Motivación 13

1.3 Objetivos 13

1.4 Organización de la memoria //revisar 13

Capítulo 2. Estado del arte 15

2.1 Análisis de aplicaciones similares 15

2.1.1 Museum webshop 15

2.1.2 Artmajeur 16

2.1.3 Artelista 17

2.2 Herramientas y tecnologías 18

2.2.1 Microsoft Office Word 18

2.2.2 Microsoft Office Project 20

2.2.3 Visual Studio Code 21

2.2.3.1 Angular 23

2.2.3.2 Strapi 24

2.2.4 MAMP 25

2.2.5 GitHub Desktop 26

2.2.6 Visual Paradigm 27

2.2.7 Adobe Photoshop 28

Capítulo 1. Planificación 30

1.1 Requisitos 30

1.2 Especificaciones 30

1.3 Costes 30

1.4 Riesgos 30

1.5 Viabilidad 30

Capítulo 2. Análisis 33

Capítulo 3. Diseño 36

Capítulo 4. Implementación 39

Capítulo 5. Pruebas 42

Capítulo 6. Conclusiones 45

Apéndice A 48

Bibliografía 51

Índice de figuras

[Figura 1: Interfaz UI de Museum Webshop 14](#_Toc108619413)

[Figura 2: Interfaz UI de Artmajeur 15](#_Toc108619414)

[Figura 3: Interfaz UI de Artelista 16](#_Toc108619415)

[Figura 4 - Captura de la interfaz UI de Libre Office 17](#_Toc108619416)

[Figura 5 – Captura de la interfaz UI del software “Eclipse” 18](#_Toc108619417)

[Figura 6 - Captura de la interfaz UI del software "Netbeans" 19](#_Toc108619418)

[Figura 7 - Captura de la interfaz UI de "Strapi" 21](#_Toc108619419)

[Figura 8 - Captura de la interfaz UI del software GitHub Desktop 22](#_Toc108619420)

[Figura 9 - Captura del software de edición fotográfica Photoshop 23](#_Toc108619421)

# Introducción

## Introducción

El arte lleva existiendo desde los tiempos prehistóricos, pero no fue hasta el Renacimiento que se comenzó el cambio a su comercialización. Durante el paso de los años, la estrategia de mercado ha ido actualizándose, para así perdurar hasta la actualidad.

Mientras que la gran mayoría de las obras clásicas más importantes están coleccionadas en museos para la contemplación de todos los ciudadanos, los artistas que desean ganarse la vida con sus obras deben ganarse la vida gracias a los encargos que reciben. Por tanto, actualmente el arte en forma de pintura sigue sobreviviendo gracias a haberse podido adaptar a las nuevas tecnologías.

La innovación en la comercialización de la pintura artística mediante internet se basa en gran medida con la aparición de las redes sociales, ya que la gran mayoría de artistas pequeños deben promocionarse a través de estas mismas para darse a conocer y así recibir encargos.

A parte, ha habido una considerable evolución con la comercialización de las pinturas debido a la aparición de las criptomonedas, ya que, con dicha aparición, se genera una nueva forma de coleccionar arte, se denomina “el cripto arte”. Entre todos los tipos de cripto arte que existen, la variante más destacada estos últimos meses es el denominado “NFT”, que vende sus obras de forma digital (imágenes, videos, etc.).

El propósito de este proyecto es crear una aplicación Web a modo de tienda de arte, enfocado en el modo de comercialización tradicional, pero apoyando a los artistas que desean ganarse la vida a través de encargos.

## Motivación

La motivación principal para hacer este TFG es poder ampliar mis conocimientos artísticos, durante el proceso de creación de una aplicación Web.

Actualmente, no se apuesta lo suficiente en el arte porque debido al mercado actual, cuesta premiar el talento. Y es por eso, que muchos creadores no pueden ganarse la vida con ello, y una de mis motivaciones es crear un espacio que se pudiera aprovechar en un futuro para ayudar a la gente a ganarse la vida con su talento.

## Objetivos

Los objetivos de este proyecto son:

* Realizar un análisis comparativo de las diferentes webs similares a mi propuesta para desarrollar una imagen de marca a la altura.
* Crear una interfaz web simple y minimalista, además de atractiva, que facilite el uso de esta a los usuarios.
* Darles visibilidad a los artistas en algún apartado de la web.
* Facilitar la administración tanto para los miembros que vayan a vender obras, como para el propio gestor de la web.

## Organización de la memoria //revisar

La memoria se ha estructurado en siete capítulos diferentes, todos juntos unen la información necesaria para plasmar todo el contenido que debe tratar este Trabajo de Fin de Grado.

El primer capítulo, que es en el que estamos, es introductorio. Se muestra sobre que va a tratar este proyecto. Además, muestra cuales son los motivos por los que se elige el tema, y los objetivos a lograr al finalizar el proyecto.

El segundo capítulo explica el estado del arte. Esto no tiene relación con el tema que tratamos. El estado del arte, según dice Nancy Piedad Molina Montoya [1], “es una modalidad de la investigación documental que permite el estudio del conocimiento acumulado dentro de un área específica”, por tanto, en este capítulo se hará un estudio de las competencias para definir el estilo de nuestra web y las tecnologías que usaremos. Además, también se define una metodología de trabajo.

En el tercer capítulo definiremos los requisitos de la aplicación, y a su vez, sus especificaciones.

# 

# Estado del arte

## Análisis de aplicaciones similares

Respecto al estado del arte, tras investigar las diferentes webshops (Tiendas Web) de arte que he ido encontrando, destaco las siguientes tres páginas, las cuales voy a analizar:

## ****Museum webshop****

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Figura 1: Interfaz UI de Museum Webshop

Si tecleamos en el buscador de Google “webshop de arte” esta página es de las primeras que aparece, si no la primera. Esto quiere decir que a nivel de posicionamiento es de las más importantes del mercado, por tanto, algo estará haciendo bien.

A nivel de diseño de interfaz, utiliza un diseño minimalista, sencillo y eficaz. Por una parte, tenemos un header, que centra toda su atención en el logotipo de la marca, y por otra parte tenemos el body (contenido), que muestra los artistas más importantes, y, por tanto, más conocidos, con sus obras más significativas. Si clicamos en alguno de los artistas nos redirecciona a la página del propio artista, con todos sus productos. El hecho de tener una interfaz de este estilo hace que para el cliente su paso por la web resulte intuitivo.

A nivel de programación, si inspeccionamos la página, hay varios elementos, como los mostrados en el apartado “fuentes”, que afirman que el lenguaje de programación usado es JavaScript. Esto nos va a servir para decidir qué lenguaje usar en nuestro propio trabajo.

Se puede acceder a esta página a través de [este enlace](https://www.museum-webshop.com/es/artistas/).

## Artmajeur

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 2: Interfaz UI de Artmajeur

Si con la anterior webshop he comentado que tenía un diseño de interfaz minimalista, con esta lo pienso el doble. Además de eso, me parece bastante más estética. Pese a mostrarse muchos menos elementos en pantalla, no deja de ser intuitiva. Como vemos por una parte está el header, con el logotipo y el menú, y por otra parte el body, con contenido destacado.

Cabe decir que esta página también está construida en JavaScript, y se puede acceder a esta a través de [este enlace](https://www.artmajeur.com/es/?gclid=CjwKCAjwsJ6TBhAIEiwAfl4TWMKCgYQv3H1mo0KdjB1R6a0c2ZQOHm7SLsejEKpsJPtd5XE7xZ5R0RoCTdYQAvD_BwE).

## Artelista

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Figura 3: Interfaz UI de Artelista

A simple vista se ve como está webshop lleva tiempo sin rediseñarse, por tener elementos en la página con estilos obsoletos.

En cuanto al diseño de la interfaz, al igual que las anteriores páginas, contiene un header con el logo, una barra de búsqueda y un menú. Sin embargo, en el body, se olvida de mostrar contenido destacado, como obras o artistas importantes (o en su defecto, más vendidas), y muestra directamente las obras recientes. Un estilo similar a páginas al que era antiguamente el de las páginas de compra y venta de cualquier tipo de artículos, como por ejemplo Milanuncios.

En cuanto a la programación, también parece estar construida en JavaScript, y se puede acceder a esta a través de [este enlace](https://www.artelista.com/?gclid=CjwKCAjwsJ6TBhAIEiwAfl4TWI0Uo4p08tLLAArPTLzlsENdlvT2gKBjjAnEnVodF40glh0ZsggU9hoCw50QAvD_BwE).

## Herramientas y tecnologías

En este apartado, se van a analizar las diferentes herramientas, además de las tecnologías, con las que se va a desarrollar el proyecto. Se van a comentar las ventajas y desventajas de cada una para explicar, porque se ha elegido esa herramienta o tecnología frente a otras.

## Microsoft Office Word

[Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente](https://www.microsoft.com/es-es/microsoft-365/free-office-online-for-the-web)

Imagen 1 - Microsoft Office Word

Microsoft Word, según Wikipedia [2], es un software de procesamiento o tratamiento de textos. Fue creado por Microsoft y está integrado por defecto en el paquete de Microsoft Office, al igual que la herramienta que veremos a continuación.

Se ha usado este software para la realización de la memoria del proyecto después de barajar entre opciones como “LibreOffice Writer”, que es el software que se muestra en la figura 4, o en su defecto, “Google Docs”. Ambos son softwares gratuitos, por lo que, por esa parte, son opciones recomendables.

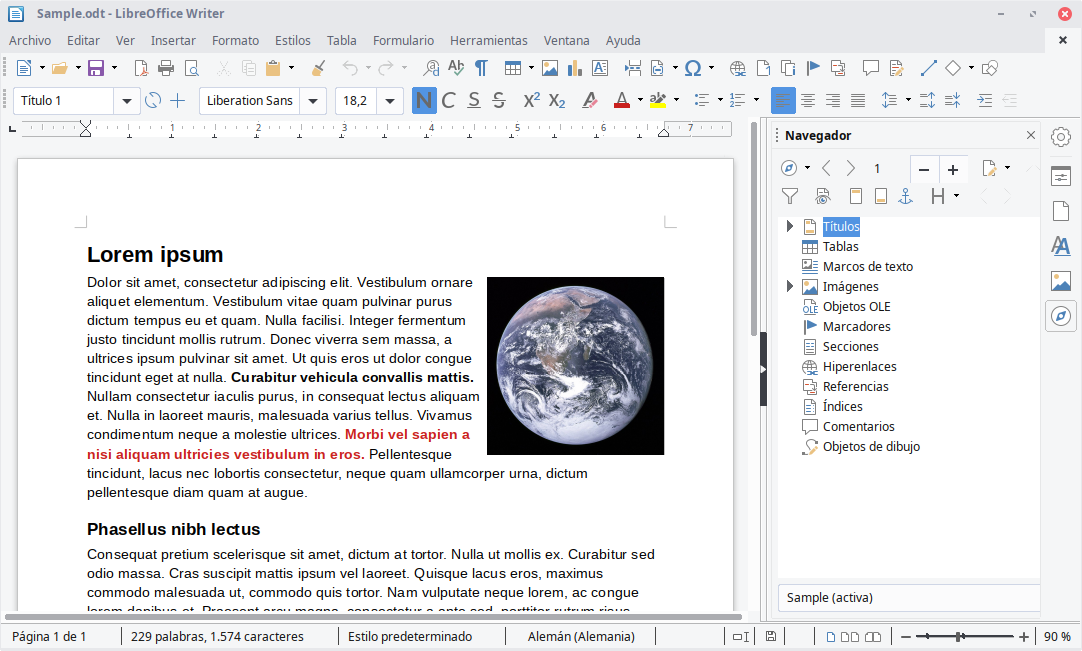


Figura 4 - Captura de la interfaz UI de Libre Office

Sin embargo, gracias a los conocimientos adquiridos en asignaturas impartidas en el grado de Ingeniería Telemática, el Microsoft Office Word es la mejor opción. Además, pese a que este software tiene la desventaja de tener que adquirir una licencia para su uso, la universidad proporciona licencia gratuita, por tanto, ese inconveniente se obvia.

## Microsoft Office Project

[Texto

Descripción generada automáticamente](https://www.microsoft.com/es-es/microsoft-365/free-office-online-for-the-web)

Imagen 2 - Microsoft Office Project

Microsoft Project, según Wikipedia [3], es un software de administración de proyectos y programas de proyectos desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.

Este software, al igual que con Microsoft Word, se elige frente a otras alternativas por la licencia y conocimientos adquiridos gracias al grado cursado en la universidad.

Microsoft Project se va a usar para realizar el apartado de la memoria del proyecto llamado “Planificación”, definiendo las actividades a realizar con su duración estimada y así crear un diagrama de Gantt.

## Visual Studio Code

[Logotipo, Icono

Descripción generada automáticamente](https://code.visualstudio.com/)

Imagen 3 - Visual Studio Code

Visual Studio Code, según sus creadores [4], es un editor de código optimizado con soporte para operaciones de desarrollo como depuración, ejecución de tareas y control de versiones.

Tras haber utilizado distintos editores de código durante el transcurso del grado, como son “Eclipse”, tal y como se muestra en la figura 5,

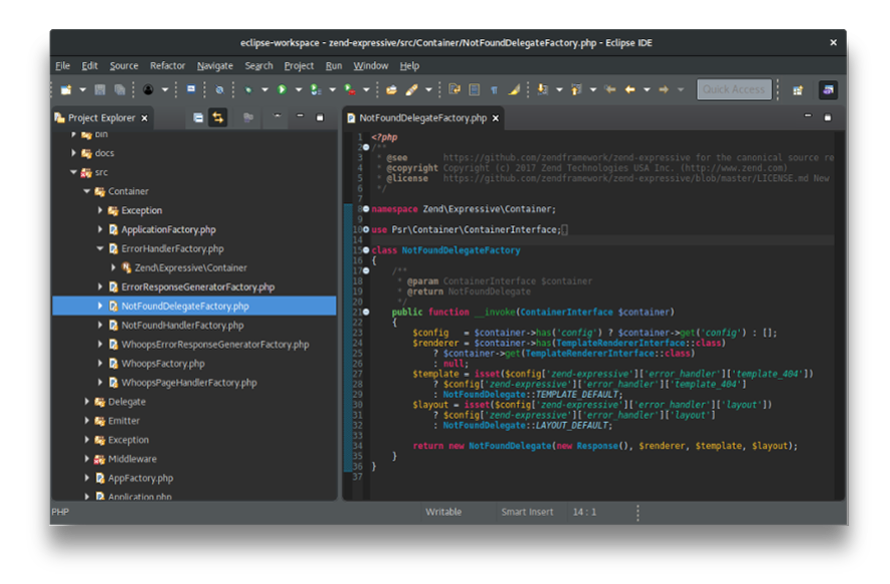


Figura 5 – Captura de la interfaz UI del software “Eclipse”

o “Netbeans”, tal y como se muestra en la figura 6, se ha decidido utilizar Visual Studio Code como software para implementar la aplicación Web. Para poder decidirse entre estos programas se han establecido una serie de requisitos que deben soportar estos editores.

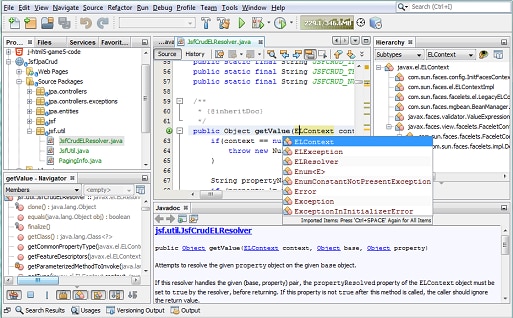


Figura 6 - Captura de la interfaz UI del software "Netbeans"

El primer requisito es poder usar Node.js, para poder instalar los paquetes necesarios para trabajar con las tecnologías deseadas, las cuales se comentarán más adelante, por tanto, también se debe poder trabajar con dichas tecnologías en estos editores. También deben tener consola de comandos para poder ejecutar ciertos comandos necesarios para lanzar la aplicación.

El segundo requisito es poder usar control de versiones con GitHub, tema del cual también se comentará más adelante.

Dado que todas las alternativas cumplen ambos requisitos, la decisión viene dada de la posibilidad que ofrece Visual Studio Code de instalar diferentes extensiones subidas por la comunidad que ayudan a hacer la programación bastante más cómoda, como puede ser, por ejemplo “Thunder Client” que sirve para hacer llamadas a una API (donde se almacenan los datos de la aplicación) y obtener su respuesta. A esta extensión se le ha dado bastante uso para hacer todo tipo de pruebas.

Gracias a las prácticas curriculares realizadas en Capgemini, una de las consultoras de software mas importantes del mundo, se han adquirido conocimientos sobre diversas tecnologías, y han sido estos conocimientos los que han decidido las tecnologías a usar, que son las siguientes:

## Angular

[Icono

Descripción generada automáticamente](https://angular.io/)

Imagen 4 - Angular

Angular, según Wikipedia [5], es un framework para aplicaciones web, desarrollado en TypeScript, un lenguaje de código abierto basado en JavaScript.

Angular proporciona ciertas características como son la velocidad y rendimiento en la generación de código, además de la productividad, al ofrecer plantillas que crean rápidamente interfaces de usuario sencillas con una sintaxis de plantilla simple.

Se podría haber realizado la aplicación Web directamente en JavaScript, pero con este framework, es más sencillo implementar una aplicación que a priori puede llegar a ser muy compleja. Esto se debe a la generación y, por ende, separación de las diferentes pantallas de la aplicación en componentes y módulos, de forma que quedan los ficheros mucho mas estructurados que con una aplicación JavaScript programada “a pelo”.

Para poder lanzar una aplicación Web, Angular trabaja con varios lenguajes:

* HTML: Es el lenguaje que se encarga de mostrarle la aplicación al usuario directamente.
* CSS: Es el lenguaje que se encarga de darle estilos a la web. Además, los estilos en este proyecto se complementan con Bootstrap, una biblioteca multiplataforma con un conjunto de diseños predefinidos para aplicaciones web.
* TypeScript: Es el lenguaje que se encarga de hacer que la aplicación funcione de manera interna. Es un lenguaje de código abierto, como se ha comentado antes, basado en JavaScript, uno de los lenguajes más usados en programación.

## Strapi

[](https://strapi.io/)

Imagen 5 - Strapi

Strapi es un CMS (Content Management System), es decir, un software que ayuda a administrar contenidos dinámicos. Antes, se ha comentado que “Thunder Client” es una extensión de Visual Studio Code que hace llamadas a una API, pues Strapi es el sistema que se encarga de generar dicha API.

Tiene una interfaz de usuario simple para que, sin conocimientos previos, se pueda crear una API de forma sencilla. En la figura 7 se muestra un ejemplo de la interfaz que ofrece Strapi. Como se puede apreciar, a la parte izquierda están los apartados a los que se puede acceder, siendo los más importantes los dos primeros. En estos apartados se gestionan tanto los tipos de datos que va a ofrecer la API, como los datos directamente.

Se ha decidido usar un CMS debido a las facilidades que ofrece frente a un Back-End programado a mano, que significaría hacer otro proyecto a parte de la misma envergadura aproximadamente del proyecto de Angular.

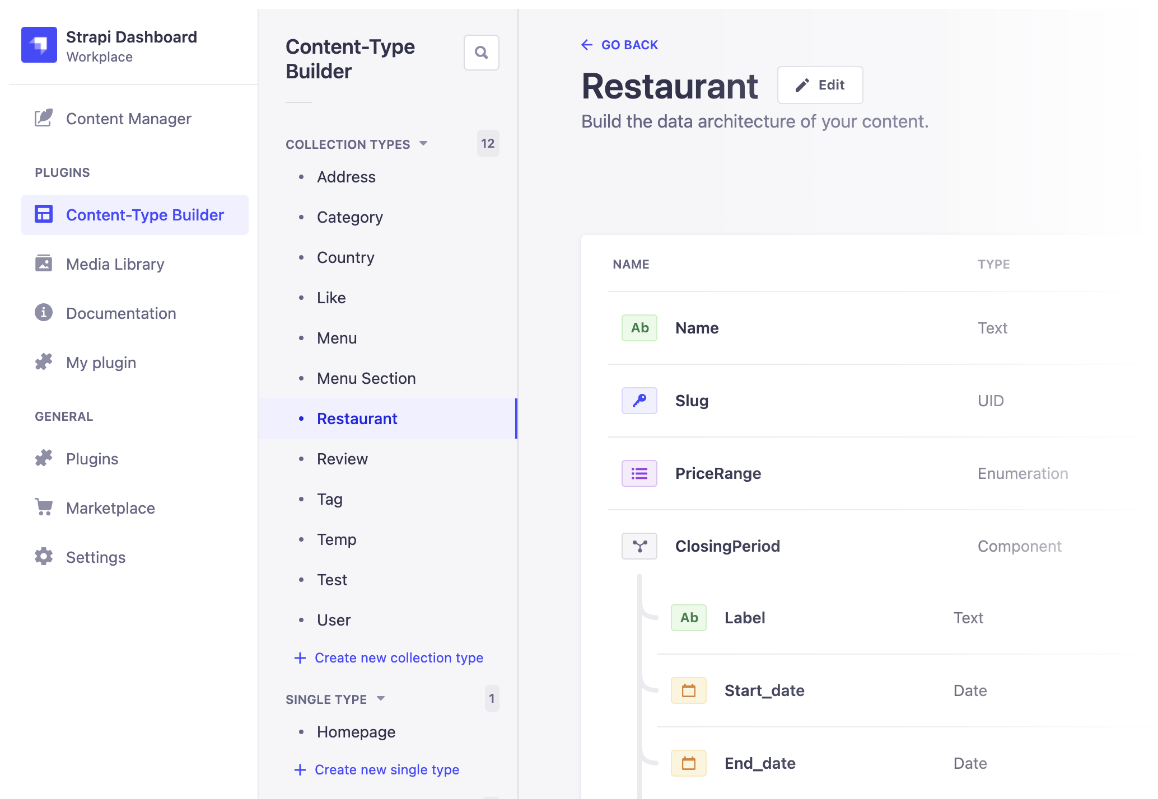


Figura 7 - Captura de la interfaz UI de "Strapi"

## MAMP

[Dibujo de un perro

Descripción generada automáticamente con confianza media](https://www.mamp.info/en/windows/)

Imagen 6 - MAMP

MAMP es un conjunto de programas software usados para desarrollar sitios web dinámicos, este en concreto sirve también para sistemas operativos Apple Macintosh. Las siglas de MAMP se refieren a MacOs, Apache (Servidor Web), MySQL (Sistema de gestión de base de datos), PHP o Python (lenguajes de programación usados).

Como MAMP, existen otras alternativas, como pueden ser XAMPP, WAMP, LAMP. Debido a que Strapi requiere de una base de datos MySQL, solo se necesita esa componente, por lo que cualquier alternativa se puede usar.

## GitHub Desktop

[Interfaz de usuario gráfica

Descripción generada automáticamente](https://desktop.github.com/)

Imagen 7 - GitHub Desktop

Como se había comentado anteriormente, se va a usar GitHub en este proyecto para gestionar el control de versiones. GitHub es un repositorio dedicado a alojar proyectos utilizando el sistema de control de versiones Git.

Se podría hacer el control de versiones desde el propio Visual Studio Code pero debido a lo intuitivo de la interfaz de la propia aplicación de escritorio de GitHub, se ha decidido usar esta para hacer estos procesos de forma más sencilla.

Texto

Descripción generada automáticamente

Figura 8 - Captura de la interfaz UI del software GitHub Desktop

Como se puede apreciar en la figura 8, las ventajas que ofrece usar la aplicación de escritorio frente a gestionar el control de versiones desde el propio editor de código, es la posibilidad de ver el historial de cambios con todas las líneas de código que se han ido modificando por cada uno de estos. Esta ventaja hace que en caso de querer volver a una versión anterior se pueda elegir con facilidad la versión que recuperar.

## Visual Paradigm

[Imagen que contiene Icono

Descripción generada automáticamente](https://www.visual-paradigm.com/)

Imagen 8 - Visual Paradigm

Visual Paradigm se va a utilizar para hacer la parte de diseño del proyecto, creando los casos de uso necesarios, y sus respectivos diagramas. Se decide usar esta herramienta (al igual que con varias herramientas comentadas anteriormente), por los conocimientos adquiridos durante el grado universitario.

Según Capterra [6], Visual Paradigm ayuda a los equipos de desarrollo de softwares a capturar los requisitos correctos y transformarlos en diseños precisos, lo que ayuda a los desarrolladores a crear el software adecuado según los requisitos. Gracias a esta posibilidad, el trabajo de implementación queda reducido al tener los algoritmos construidos en forma de diagramas.

## Adobe Photoshop

[Icono

Descripción generada automáticamente](https://www.adobe.com/es/products/photoshop/landpb.html?mv=search&mv=search&sdid=LZ32SYVR&ef_id=CjwKCAjw_b6WBhAQEiwAp4HyIHs_oZCzG7sNfgO1ShSSn076nVB67KurgnFU1LeSgiurJM5u68T-fBoCoh0QAvD_BwE:G:s&s_kwcid=AL!3085!3!441887029672!e!!g!!adobe%20photoshop!1445901735!56657232416&gclid=CjwKCAjw_b6WBhAQEiwAp4HyIHs_oZCzG7sNfgO1ShSSn076nVB67KurgnFU1LeSgiurJM5u68T-fBoCoh0QAvD_BwE)

Imagen 9 - Adobe Photoshop

Finalmente, se ha decidido usar Adobe Photoshop, uno de los programas de edición fotográfica mas usados en la actualidad, para realizar la estética de la aplicación, creando así la iconografía, la paleta de colores, y los tipos de fuente a utilizar, de modo que, en la parte de diseño del proyecto se mostrarán los diseños realizados.

Interfaz de usuario gráfica

Descripción generada automáticamente

Figura 9 - Captura del software de edición fotográfica Photoshop

Se ha decidido principalmente este software frente a otros por, tal y como se ve en la figura 9, la posibilidad de encapsular los elementos en mesas de trabajo. Además, ofrece muchas herramientas para vectorizar el contenido y así exportarlo en una imagen “.svg”, de forma que no pierda calidad pese a aumentar o reducir el tamaño de esta. Esta propiedad en páginas web es muy importante para una estética limpia.

# Planificación

## Requisitos

## Especificaciones

## Costes

## Riesgos

## Viabilidad

# Análisis

# Diseño

# Implementación

# Pruebas

# Conclusiones

# Apéndice A

# Bibliografía

|  |  |
| --- | --- |
| [1] | N. P. M. Montoya, «Ciencia y Tecnología para la Salud Visual y Ocular,» 24 Octubre 2005. [En línea]. Available: https://ciencia.lasalle.edu.co/svo/vol3/iss5/10/. |
| [2] | Wikipedia, «Microsoft Word,» [En línea]. Available: https://es.wikipedia.org/wiki/Microsoft\_Word. |
| [3] | Wikipedia, «Microsoft Project,» [En línea]. Available: https://es.wikipedia.org/wiki/Microsoft\_Project. |