

Grado en Ingeniería

Texto

Descripción generada automáticamenteTelemática

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

Índice general 9

Índice de figuras 11

Capítulo 1 – Introducción 12

Introducción 12

Motivación 13

Objetivos 13

Organización de la memoria 13

Capítulo 2 - Estado del arte 16

Análisis de aplicaciones similares 16

Museum webshop 16

Artmajeur 16

Artelista 16

Herramientas y tecnologías 16

Microsoft Office Word 16

Microsoft Office Project 16

Visual Studio Code 16

Angular 19

Strapi 20

MAMP 21

GitHub Desktop 22

Visual Paradigm 23

Adobe Photoshop 24

Capítulo 3 - Planificación 26

Requisitos 26

Requisitos funcionales 26

Requisitos no funcionales 28

Planificación temporal 30

Costes 33

Riesgos 33

Viabilidad 33

Capítulo 4 – Análisis 36

Capítulo 5 - Diseño 39

Capítulo 6 - Implementación 42

Capítulo 7 - Pruebas 45

Capítulo 8 - Conclusiones 48

Apéndice A 51

Bibliografía 54

Índice de figuras

[Figura 1: Interfaz UI de Museum Webshop 15](#_Toc108619413)

[Figura 2: Interfaz UI de Artmajeur 16](#_Toc108619414)

[Figura 3: Interfaz UI de Artelista 17](#_Toc108619415)

[Figura 4 - Captura de la interfaz UI de Libre Office 19](#_Toc108619416)

[Figura 5 – Captura de la interfaz UI del software “Eclipse” 21](#_Toc108619417)

[Figura 6 - Captura de la interfaz UI del software "Netbeans" 22](#_Toc108619418)

[Figura 7 - Captura de la interfaz UI de "Strapi" 24](#_Toc108619419)

[Figura 8 - Captura de la interfaz UI del software GitHub Desktop 26](#_Toc108619420)

[Figura 9 - Captura del software de edición fotográfica Photoshop 28](#_Toc108619421)

Capítulo 1 – 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 conocimientos artísticos, durante el proceso de creación de una aplicación Web. Para poder realizar este proyecto, se va a tener que realizar un trabajo de investigación tanto sobre las aplicaciones similares, como del estado actual del arte y sus necesidades, para así formar una propuesta. De este proceso nace la motivación principal.

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 las motivaciones consiste en crear un espacio que se pudiera aprovechar en un futuro para ayudar a la gente a ganarse la vida con su talento.

Otra motivación es poder aplicar los conocimientos aprendidos en programación *Front End* durante el transcurso del grado universitario para crear una aplicación estética y con una personalidad que corresponda a las necesidades de esta. Debido a que se va a crear una aplicación sobre venta de arte, se plantea enfocar gran parte del proyecto en crear una aplicación que se vea como tal.

Objetivos

El objetivo principal de este proyecto es crear una aplicación que tanto los posibles clientes como artistas, se sientan cómodos de usar la web. El objetivo es conseguir esto de dos formas:

* Centrando la estética de esta en el propio contenido que van a subir los artistas (como si de una *webshop* (Tienda Web) real se tratase).
* Facilitando la administración a todo tipo de usuario, ya sean clientes, artistas o administradores.

Otro objetivo importante es que los artistas obtengan el reconocimiento que se merecen, para ello se pretende que, en algún apartado de la web se les de visibilidad.

Para garantizar estos objetivos, se pretende realizar un estudio exhaustivo de, como se ha comentado anteriormente, las aplicaciones similares que ofrecen este tipo de servicios para así ofrecer ya sea la misma calidad de servicio o incluso mejor.

Finalmente, un objetivo más tapado consiste en aprender de todo el proceso de realizar el proyecto, para no cometer errores posteriormente que pudiesen darse durante el transcurso del mismo.

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.

Capítulo 2 - Estado del arte

Análisis de aplicaciones similares

Respecto al estado del arte, tras investigar las diferentes *webshops* de arte que se han ido encontrando, se destacan las siguientes tres páginas:

Museum webshop

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Figura : Interfaz UI/UX de Museum Webshop

Si se teclea en el buscador de Google el término “*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.

A nivel de diseño de interfaz, utiliza un diseño minimalista, sencillo y eficaz.

Por una parte, existe un *header* (encabezado), que centra toda su atención en el logotipo de la marca, teniendo también como elementos las distintas subpáginas a las que se puede acceder, además del carrito, un cuadro de búsqueda, y opciones para inicio de sesión y demás.

Por otra parte, la página tiene un *body* (contenido), que muestra los artistas más importantes, y, por tanto, más conocidos, con sus obras más significativas. Si se cliquea 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.

Pese a tener bastante información en la página principal, los elementos que no son el contenido principal de la página (las obras a comprar), no opacan demasiado el propio contenido. Esta opción consigue tener bastantes elementos en pantalla de modo que el usuario tenga que hacer pocos *clicks* para llegar a los sitios, además de captar la atención de las obras.

A nivel de programación, no se conoce mucha información, solo que al inspeccionar la página, hay varios elementos, como los mostrados en el apartado “fuentes”, con *.js* como tipo de dato. Esto significa que se ha construido la aplicación con una serie de ficheros JavaScript para su despliegue. Realmente no dice gran cosa además de verificar que se ha hecho en un lenguaje posible de construir a JavaScript para su despliegue

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 : Interfaz UI/UX de Artmajeur

Esta web se distingue principalmente de la anterior al tener una interfaz mucho más minimalista.

Por una parte, el *header* pese a tener un numero aproximado de elementos al anterior, da la sensación de que tiene bastantes menos, esto se debe a los espacios que hay entre elementos. Genera una interfaz bastante mas limpia sin romper apenas la funcionalidad de la aplicación.

Por otra parte, el *body* cambia considerablemente respecto a la página anterior. En esta se ve como primer elemento una imagen con disposición horizontal que ocupa el ancho completo de la página. Esta característica se usa para centrar toda la atención del usuario en la propia imagen. Como se puede apreciar, la imagen sirve de navegación a una subpágina de la web (al igual que con los artistas en la web anterior), y se suelen poner las subpáginas en tendencia como portada, es decir, las que existen solo para días especiales o para las obras mas vendidas por ejemplo.

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 : Interfaz UI/UX de Artelista

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

En cuanto al diseño de la interfaz, al igual que las anteriores páginas, contiene un *header* con los elementos propios de una página de estas características. Sin embargo, pese a tener pocos elementos, debido a los pocos espacios que hay, da sensación de sobrecarga.

Sin embargo, en el *body* se muestran directamente las últimas obras con sus precios. Con este estilo no se consigue captar la atención del usuario tanto como en las anteriores páginas. Contiene un estilo similar al que tenían antiguamente 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 - 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.

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.

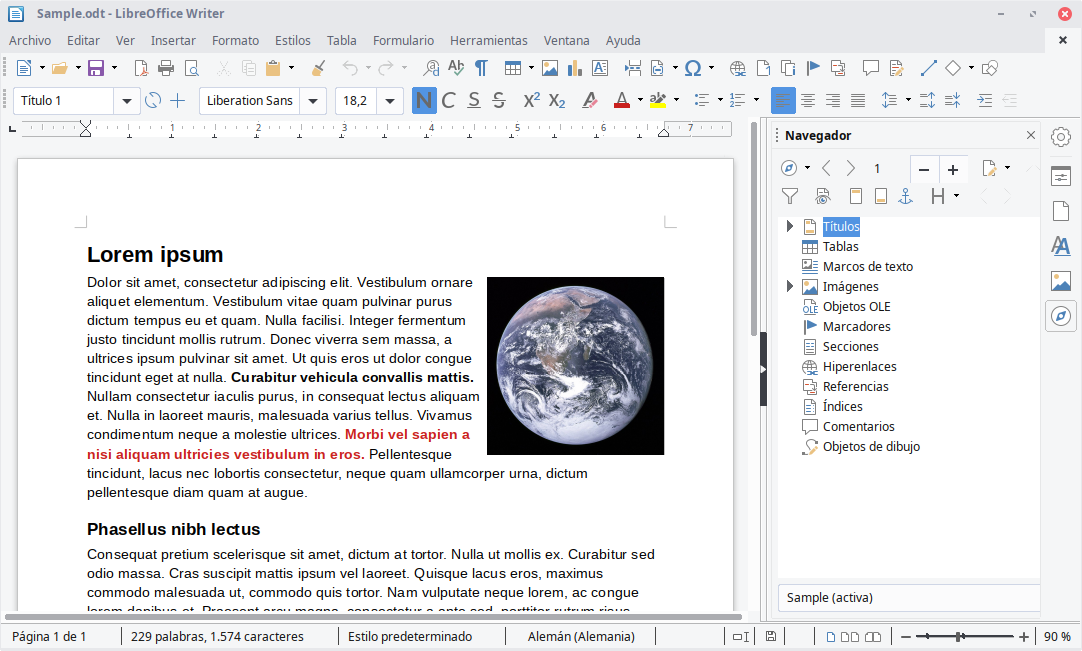


Figura - Captura de la interfaz UI de Libre Office

Microsoft Office Project

[Texto

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

Imagen - 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 - 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,

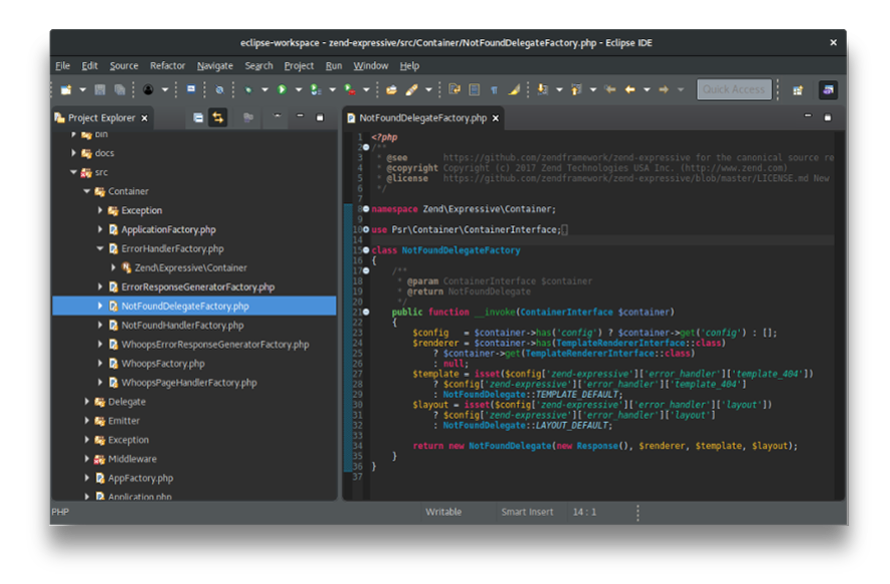


Figura – Captura de la interfaz UI del software “Eclipse”

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.

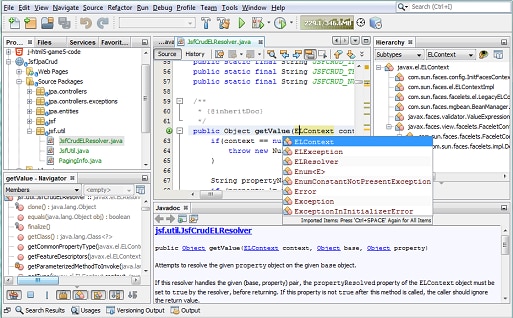


Figura - Captura de la interfaz UI del software "Netbeans"

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.

Como se va a comentar posteriormente, se van a realizar dos proyectos, uno de Angular y otro de Strapi. Ambos hacen uso de node.js para su desarrollo. Por ejemplo, el comando *npm i* instala el directorio *node\_modules* en ambos proyectos, el cual es obligatorio tener para que funcione la aplicación. Además, Strapi hace uso del comando *npm run develop* para el lanzamiento de su proyecto. Por tanto, queda demostrado que el editor utilizado debe tener compatibilidad con Node.js para poder realizar el proyecto. Aunque se podría trabajar con una consola de comandos externa igualmente, se busca la mayor comodidad posible.

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 más 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 - 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 más 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 - 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.

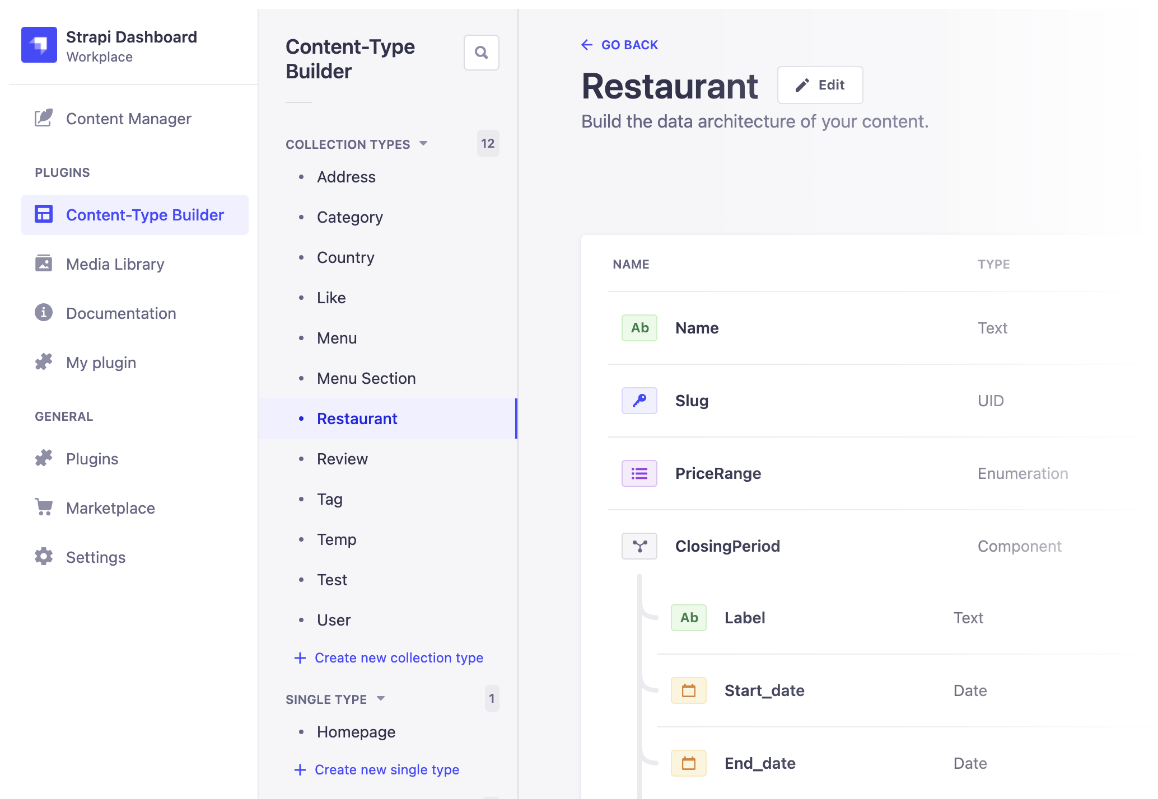


Figura - Captura de la interfaz UI de "Strapi"

MAMP

[Dibujo de un perro

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

Imagen - 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.

En este proyecto se ha decidido usar MAMP principalmente por las facilidades en la configuración que ofrece. Para usarlo en primera instancia solo hay que abrir el programa y automáticamente genera el servidor con la base de datos.

GitHub Desktop

[Interfaz de usuario gráfica

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

Imagen - 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 - 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 - 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.

Este *software* ofrece bastantes facilidades para el diseño de un proyecto, pudiendo crear fácilmente los diagramas que se necesitan para este, los cuales son:

* Diagrama de Casos de Uso
* Diagrama de clases del modelo de datos
* DSGS

Finalmente, Visual Paradigm ofrece también la posibilidad de gestionar el control de versiones desde la propia interfaz del programa. Para un proyecto con un volumen como este es importante que los *softwares* utilizados ofrezcan características como el control de versiones dado el número de cambios que se van a producir durante el proceso.

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 - Adobe Photoshop

Finalmente, se ha decidido usar Adobe Photoshop, uno de los programas de edición fotográfica más 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 - 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.

Capítulo 3 - Planificación

Requisitos

Los requisitos de una aplicación son fundamentales para la realización correcta de un proyecto. En este apartado se van a detallar los requisitos de la aplicación, distinguiendo entre dos tipos: funcionales y no funcionales. Los requisitos funcionales especifican “que” debe hacer el sistema, mientras que los requisitos no funcionales especifican “como” debe hacerlo.

Requisitos funcionales

1. Al acceder a la página principal, el sistema deberá mostrar un listado de los últimos tres productos de cada categoría, siendo estas: Accesorios, Figuras y Pinturas.
2. Además de la página principal, el sistema debe permitir al usuario acceder a las propias páginas de cada categoría, y a la página de artistas.
3. Al acceder a la página de Accesorios, el sistema debe permitir al usuario ver los productos que tienen como tipo de producto el valor “Accesorio”. El usuario podrá acceder a la información detallada de cada producto y agregarlo a la bolsita. El usuario debe tener la opción de filtrar los productos por tipo de accesorio y nombre.
4. Al acceder a la página de Accesorios, el sistema debe permitir al usuario ver los productos que tienen como tipo de producto el valor “Accesorio”. El usuario podrá acceder a la información detallada de cada producto y agregarlo a la bolsita. El usuario debe tener la opción de filtrar los productos por tipo de accesorio y nombre.
5. Al acceder a la página de Figuras, el sistema debe permitir al usuario ver los productos que tienen como tipo de producto el valor “Figura”. El usuario podrá acceder a la información detallada de cada producto y agregarlo a la bolsita. El usuario debe tener la opción de filtrar los productos por tipo de figura y nombre.
6. Al acceder a la página de Pinturas, el sistema debe permitir al usuario ver los productos que tienen como tipo de producto el valor “Pintura”. El usuario podrá acceder a la información detallada de cada producto y agregarlo a la bolsita. El usuario debe tener la opción de filtrar los productos por tipo de pintura y nombre.
7. Al acceder a la página de Artistas, el sistema debe permitir al usuario ver los usuarios con rol “Artista” registrados en la base de datos y acceder a su información detallada, dentro de esta, el usuario tendrá la opción de acceder a sus productos. El sistema debe permitir al usuario ver los productos de dicho artista, acceder a su información detallada y agregarlos a la bolsita.
8. El sistema debe permitir al usuario iniciar sesión en la aplicación si está registrado en la base de datos, debiendo introducir su usuario y contraseña. Una vez el usuario tiene la sesión iniciada, el sistema debe permitir al usuario ver un menú con las opciones: “Editar perfil”, “Mis pedidos”, “Perfil de artista” y “Cerrar Sesión”
9. El sistema debe permitir registrar un usuario en la base de datos. Para ello se deberá proporcionar un nombre de usuario, un email, y una contraseña. La contraseña además de requerir un mínimo de 8 caracteres, deberá escribirse dos veces para verificar que se ha escrito bien. Todos los campos deberán ser obligatorios. Al registrar el usuario, este obtendrá el rol de “Cliente”, y se guardará en la base de datos.
10. El sistema debe permitir añadir productos a la bolsita a través de cualquier categoría sin necesidad de tener una sesión iniciada. Pero para realizar un pedido sobre esa bolsita, será obligatorio tener una sesión iniciada. El pedido contiene detalles de la bolsita, la información de envío, y el método de pago. En cuanto se rellenan todos los campos sin dejar ninguno vacío, se podrá realizar el pedido y este se guardará en la base de datos.
11. Al acceder a la página ”Editar perfil”, el sistema debe permitir al usuario identificado ver su información de usuario, y editarla. Podrá cambiar todos los campos, incluyendo su avatar, excepto el email, que se quedará como campo bloqueado.
12. Al acceder a la página ”Mis pedidos”, el sistema debe permitir al usuario identificado ver la información detallada de cada uno de sus pedidos realizados y podrá cancelarlos siempre y cuando dichos pedidos estén en el estado “Procesando” o “Enviado”.
13. Al acceder a la página “Perfil de artista”, el sistema deberá comprobar que rol tiene el usuario identificado. Si el usuario tiene como rol “Artista” o “Administrador” podrá ver la información de artista, la cual son los productos enviados por dicho usuario, de los cuales podrá editar su información. Además, podrá crear un producto nuevo, debiendo introducir como campos obligatorios el título, el precio, y la categoría de producto (Accesorio, Pintura, Figura); como campos opcionales podrá introducir una imagen del producto, y una descripción de este. El precio debe tener un valor mayor que 0. Cuando se cumplen todos los requisitos en los campos introducidos, se podrá crear (o editar) el producto, y se guardará su información en la base de datos. Si el usuario tiene como rol “Cliente”, no podrá ver esta información, pero tendrá la opción de solicitar los privilegios de “Artista”.
14. El sistema debe permitir al usuario con rol “Administrador” acceder al perfil de administrador, que aparecerá como una opción más en el menú del usuario identificado. Este perfil tendrá como opciones acceder a los usuarios registrados, donde podrá eliminar los que quiera; a los pedidos, donde podrá cambiar su estado, siendo las posibles opciones: “Procesando”, “Enviado”, “Recibido”, “Cancelado”; y a las solicitudes de artista, donde podrá y rechazarlas o aprobarlas para cambiar los roles de los usuarios a “Artista” según se decida.
15. El sistema debe permitir al usuario identificado cerrar la sesión cuando así lo desee.

Requisitos no funcionales

* + - 1. Todos los datos serán almacenados en una base de datos de MySQL
      2. Las tablas de la base de datos deben crearse con el CMS Strapi.
      3. La aplicación debe programarse en Angular, de modo que los lenguajes usados serán HTML, CSS y TypeScript.
      4. La interfaz de la aplicación debe realizarse de modo que sea lo mas intuitiva posible, para que sea sencilla para cualquier público.

Planificación temporal

Debido al groso de actividades que implica este proyecto, es obligatorio definir todas las actividades y agruparlas en bloques. Con esto se consigue facilitar el control de la organización de proyecto, pudiendo planificar temporalmente estas actividades creando dependencia de unas sobre otras.

El objetivo de planificar temporalmente un proyecto es definir todas las tareas determinando su duración de modo que se pueda detectar de una manera simple las que actividades críticas y, por tanto, las que pueden ocasionar más retrasos.

Para poder estimar la duración de cada actividad se ha usado la estimación por 3 valores siendo estos:

* ***to***  = Tiempo más optimista
* ***tm*** = Tiempo más probable
* ***tp***  = Tiempo más pesimista

Mediante estos 3 valores se calcula el tiempo estimado (**te**), mediante en la siguiente fórmula (Figura 10), en la que se da el mayor peso al tiempo más probable:

Figura – Formula estimación por 3 valores

Una vez definidas todas las actividades y calculadas las duraciones, queda definir las dependencias. Una actividad es dependiente de otra cuando necesita que esta acabe, para poder completarse. Esto sucede, por ejemplo, cuando una actividad requiere datos que se consiguen en otra, por tanto, hasta que esta no se complete no podrá avanzar.

Como todas las actividades las va a realizar una misma persona (aunque en cada fase del proyecto adopte distintos roles), estas no se pueden solapar, y por tanto, deben seguir un orden secuencial de modo que todas las actividades sean dependientes de las anteriores, independientemente de que tengan otras dependencias.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tarea | Nombre de la tarea | *to* [días] | *tm* [días] | *tp* [días] | *te* [días] |
| 1 | **SellArt** | | | | |
| 2 | **Análisis** | | | | |
| 3 | Análisis del estado del arte | *1.5* | *2* | *3* | *2.083* |
| 4 | Requisitos del proyecto | *1.5* | *2* | *3* | *2.083* |
| 5 | Costes del proyecto | *2.5* | *3* | *4* | *3.083* |
| 6 | Gestión de riesgos y viabilidad | *1.5* | *2* | *3* | *2.083* |
| 7 | Casos de Uso y DSGSs | *4* | *5* | *6.5* | *5.083* |
| 8 | **Diseño** |  |  |  |  |
| 9 | Modelado de la base de datos | *2.5* | *3* | *4* | *3.083* |
| 10 | Diseño de la aplicación Web | *4* | *5* | *6.5* | *5.083* |
| 11 | **Implementación** | | | | |
| 12 | Interfaz de usuario | *12* | *15* | *17* | *14.833* |
| 13 | Base de datos | *1.5* | *2* | *3* | *2.083* |
| 14 | Aplicación | *25* | *30* | *32* | *29.5* |
| 15 | **Pruebas y detección de errores** | | | | |
| 16 | Testing | *4* | *5* | *6.5* | *5.083* |
| 17 | Correción | *4* | *5* | *6.5* | *5.083* |
| 18 | Redactación de la memoria | *64* | *79* | *95* | *79.167* |

Tabla – Actividades estimadas temporalmente

Una vez hecho el calculo de la duración estimada de cada actividad (Tabla 1), se han definido las actividades en Microsoft Office Project, con sus duraciones y dependencias correspondientes. Posteriormente se ha creado el diagrama de Gantt, el cual se representa en la Figura 11.

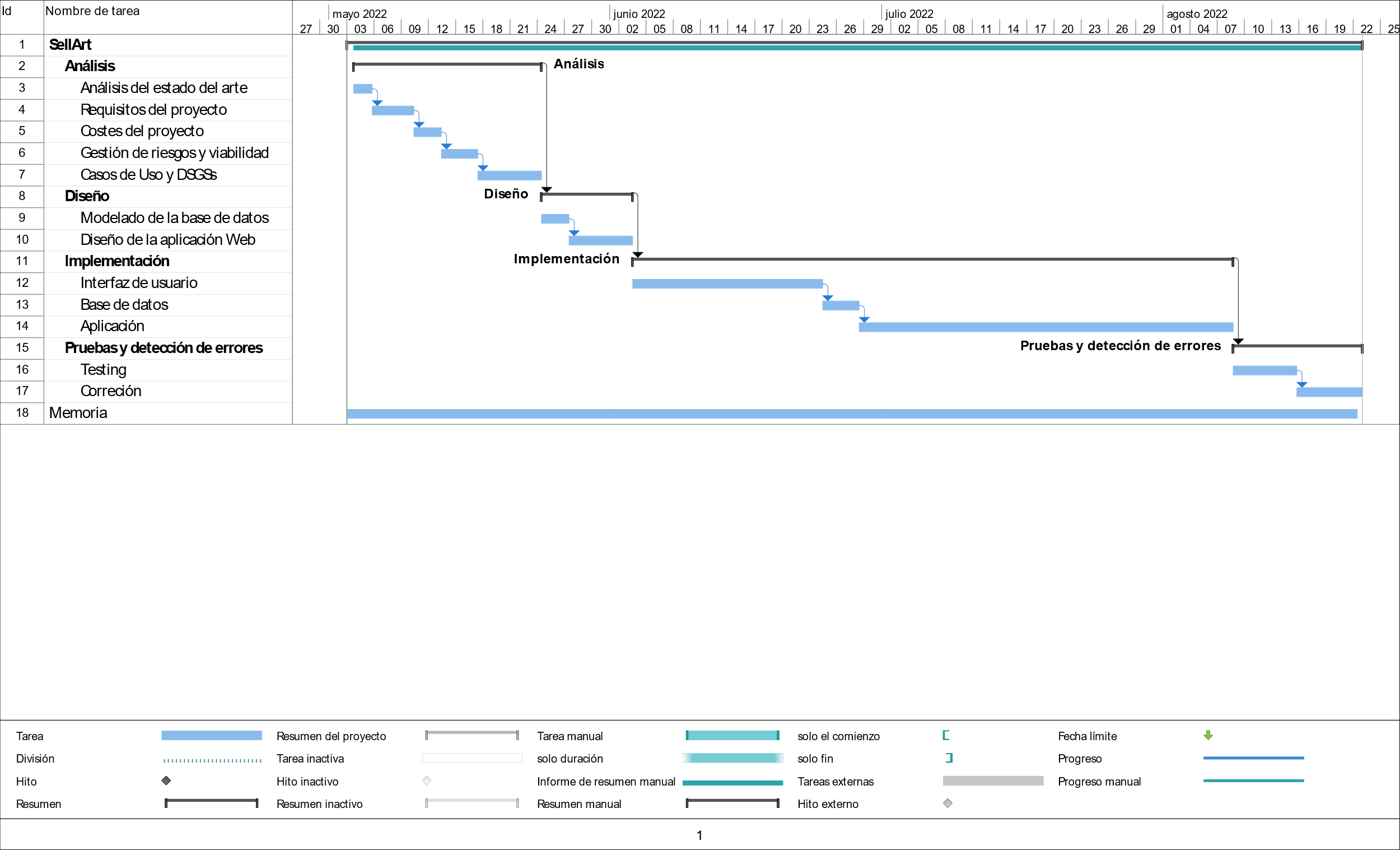


Figura – Diagrama de Gantt

Costes

Para definir los costes, en primer lugar, se necesita información sobre los sueldos de cada empleado. Pese a que en el proyecto va a trabajar únicamente una persona, esta adopta distintos roles, y, por ende, debería cobrar el sueldo proporcional por cada rol que adopte.

Como se ha visto en la definición de actividades (Tabla 1), en este proyecto existen cuatro roles distintos, es decir, cuatro empleados con distintos cargos:

* Analista
* Diseñador
* Programador
* Ingeniero de pruebas

Debido a no existir ninguna web que asegure un sueldo preciso para cada uno de los puestos requeridos se va a contrastar la información entre estas webs: “TuSalario [7]” y “Glassdoor [8]”.

**Salarios medios:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | *Salario medio [€/año]* | | |
| Empleo | **“TuSalario”** | **“Glassdoor”** | **Estimado** |
| Analista | 29.184 | 26.824 | 28.004 |
| Diseñador | 14.928 | 17.564 | 16.246 |
| Programador | 20.712 | 20.872 | 20.792 |
| Ingeniero de pruebas | 25.416 | 27.752 | 26.584 |

Tabla – Estimación salarios medios

Una vez estimados los salarios medios anuales (Tabla 2), se han introducido como empleados en el proyecto de Microsoft Office Project, tal y como se muestra en la Figura 12, para calcular el coste para cada tarea.

Tabla

Descripción generada automáticamente

Figura – Creación de recursos en M. Project

Como aproximadamente se van a dedicar 4 horas al día al proyecto, se ha de definir esta jornada en el proyecto para calcular los costes como es debido.

Además de los recursos de los empleados, es necesario también especificar los costes de material, y los gastos indirectos:

* Para los costes de material, hay que contar tanto con las licencias necesarias para el desarrollo de la aplicación (MySQL), como con las herramientas necesarias para poder producirla (equipo informático).
* Los gastos indirectos son los costes de mantenimiento de lugar donde se realiza el proyecto, es decir, luz, agua, internet, etc.

La creación de esos costes se muestra en la Figura 13, y una vez creados se asignan tal y como se muestra en la Figura 14.

Tabla

Descripción generada automáticamente

Figura – Creación de costes

Tabla

Descripción generada automáticamente

Figura – Asignación de costes

Una vez todos tanto los recursos como los costes están asignados, visualizando las estadísticas en la información general del proyecto, se obtiene el coste estimado del mismo, tal y como se muestra en la Figura 15.

Tabla

Descripción generada automáticamente

Figura – Coste estimado del proyecto

Riesgos

Según el libro proporcionado en la asignatura de Gestión de Proyectos [9], un riesgo de un proyecto es un evento o condición inciertos que, si se produce, tiene un efecto positivo o negativo sobre al menos un objetivo del proyecto, como tiempo, coste, alcance o calidad.

Los procesos para seguir una gestión de riesgos de un proyecto son los siguientes:

* Identificación de riesgos
* Selección de riegos
* Mitigación de riesgos
* Contingencia de riesgos

Identificación de riesgos

Para realizar una correcta identificación de riesgos se deben distinguir los riesgos entre ciertas características, de modo que habrá que definir tanto la probabilidad de riesgo, como su impacto, para posteriormente calcular el nivel de riesgo y así poder catalogarlos.

* La probabilidad de riesgo se distingue entre porcentajes (0 – 100%)
* El impacto de riesgo se distingue entre valores numéricos (0 – 20)

Al identificar tanto la probabilidad como el impacto de un riesgo, se calcula su nivel de riesgo, multiplicando el porcentaje de la probabilidad por el valor del impacto.

Se van a identificar los posibles riesgos asumiendo todas las partes de un proyecto, desde la parte de la planificación hasta la parte de la elaboración de pruebas.

En la Tabla 3 están contenidos los riesgos que tiene este proyecto.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Código de riesgo** | | **Prob.** | **Impacto** | **Riesgo** |
| **A** | **Elaboración de la Planificación** | **Máximo** | | **8,8** |
| A.1 | Planificación optimista, en lugar de realista. | 70% | 10 | 7,0 |
| A.2 | La planificación no incluye tareas necesarias. | 80% | 10 | 8,0 |
| A.3 | Un retraso en una tarea produce retrasos en cascada en las tareas dependientes. | 80% | 11 | 8,8 |
| A.4 | Las áreas desconocidas del producto llevan más tiempo del esperado en diseño e implementación | 60% | 7 | 4,2 |
| **B** | **Organización y Gestión** | **Máximo** | | **8,8** |
| B.1 | La planificación es demasiado mala para ajustarse a la velocidad de desarrollo deseada. | 50% | 12 | 6,0 |
| B.2 | Los planes del proyecto se abandonan por la presión, llevando al caos y a un desarrollo ineficiente. | 30% | 15 | 4,5 |
| B.3 | Se toman decisiones que alargan la planificación. | 80% | 11 | 8,8 |
| **C** | **Ambiente/Infraestructura de Desarrollo** | **Máximo** | | **10,5** |
| C.1 | Se insiste en nuevos requisitos. | 70% | 15 | 10,5 |
| C.2 | En el último momento al usuario no le gusta el producto, por lo que hay que volver a diseñarlo y construirlo. | 50% | 20 | 10 |
| C.3 | Las herramientas de desarrollo no están disponibles en el momento deseado. | 80% | 7 | 5,6 |
| **D** | **Requisitos** | **Máximo** | | **11,2** |
| D.1 | Los requisitos no se han definido correctamente, y su redefinición aumenta el ámbito del proyecto. | 80% | 13 | 10,4 |
| D.2 | Se añaden requisitos extra. | 70% | 16 | 11,2 |
| D.3 | Las partes del proyecto que no se han especificado claramente consumen más tiempo del esperado. | 90% | 12 | 10,8 |
| **E** | **Producto** | **Máximo** | | **12,0** |
| E.1 | El trabajo con un entorno software desconocido causa problemas imprevistos. | 80% | 15 | 12,0 |
| E.2 | El trabajo con un entorno hardware desconocido causa problemas imprevistos. | 70% | 15 | 10,5 |
| **F** | **Diseño e Implementación** | **Máximo** | | **9,0** |
| F.1 | Un diseño demasiado sencillo no cubre las cuestiones principales, con lo que hay que volver a diseñar e implementar. | 50% | 15 | 7,5 |
| F.2 | Un mal diseño implica volver a implementar. | 60% | 15 | 9,0 |

Tabla – Identificación de riesgos

Selección de riesgos

En esta sección se seleccionan los riesgos más críticos por categoría, y de los que tengan un mayor nivel de riesgo se recomienda una acción, ya sea de mitigación o de contingencia. En la Tabla 4 se muestra este proceso.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Código de riesgo** | | **Prob.** | **Impacto** | **Riesgo** | **Acc. Recomen** |
| **A** | **Elaboración de la Planificación** | | | | |
| A.3 | Un retraso en una tarea produce retrasos en cascada en las tareas dependientes. | 80% | 11 | 8,8 |  |
| **B** | **Organización y Gestión** | | | | |
| B.3 | Se toman decisiones que alargan la planificación. | 80% | 11 | 8,8 |  |
| **C** | **Ambiente/Infraestructura de Desarrollo** | | | | |
| C.1 | Se insiste en nuevos requisitos. | 70% | 15 | 10,5 | **Contingencia** |
| **D** | **Requisitos** | | | | |
| D.2 | Se añaden requisitos extra. | 70% | 16 | 11,2 | **Mitigación** |
| **E** | **Producto** | | | | |
| E.1 | El trabajo con un entorno software desconocido causa problemas imprevistos. | 80% | 15 | 12,0 | **Mitigación** |
| **F** | **Diseño e Implementación** | | | | |
| F.2 | Un mal diseño implica volver a implementar. | 60% | 15 | 9,0 |  |

Tabla - Selección de riesgos

Contingencia de riesgos

Tanto en este apartado como en el siguiente se indican las medidas a tomar según los riesgos seleccionados en el anterior apartado. En la tabla 5 se muestran dichas medidas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Código de riesgo** | | **Prob.** | **Impacto** | **Riesgo** |
| C.1 | Se insiste en nuevos requisitos. | 70% | 15 | 10,5 |
| **Medidas de contingencia** | | | | |
| 1 | Aumentar el plazo de finalización del proyecto. | | | |
| 2 | No aceptar los requisitos y seguir con el desarrollo de los requisitos acordados en la planificación del proyecto. | | | |

Tabla – Medidas de contingencia para el riesgo C.1

Mitigación de riesgos

Al igual que en apartado anterior, en este se muestran las medidas a tomar según los riesgos seleccionados. En las tablas 6 y 7 se muestran dichas medidas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Código de riesgo** | | **Prob.** | **Impacto** | **Riesgo** |
| D.2 | Se añaden requisitos extra. | 70% | 16 | 11,2 |
| **Medidas de contingencia** | | **Nueva prob.** | **Nuevo impacto** | **Nuevo riesgo** |
| 1 | Especificar en la planificación que solo se va a hacer los requisitos asumidos en primera instancia. | 50% | 10 | 5,0 |
| 2 | Acordar que añadir requisitos implica aumentar el plazo de finalización del proyecto. |
| 3 | Planificar un intervalo de tiempo desde que se acaba el producto, hasta que se lanza, para añadir nuevos requisitos si fuese necesario. |

Tabla – Medidas de mitigación para el riesgo D.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Código de riesgo** | | **Prob.** | **Impacto** | **Riesgo** |
| E.1 | El trabajo con un entorno software desconocido causa problemas imprevistos. | 80% | 15 | 12,0 |
| **Medidas de contingencia** | | **Nueva prob.** | **Nuevo impacto** | **Nuevo riesgo** |
| 1 | Utilizar un entorno software con el que se haya trabajado anteriormente. | 20% | 3 | 0,6 |
| 2 | Hacer un cursillo informativo sobre el software antes de su uso. |
| 3 | Hablar con especialistas en dicho software para ampliar conocimientos y minimizar riesgos. |

Tabla – Medidas de mitigación para el riesgo E.1

Viabilidad

En este apartado se hace un estudio sobre si es viable hacer este proyecto. Se va a distinguir este estudio en los siguientes tres apartados:

Viabilidad económica

El estudio sobre la viabilidad económica trata sobre el estudio de si se van a obtener beneficios y en caso de que sí, comentar como hacerlo.

En el caso de este proyecto, tal y como se comenta en apartados anteriores, el objetivo principal es crear una *webshop* de arte, de forma que la propia aplicación sea la intermediaria. Por tanto, por cada venta, la web se podría llevar un pequeño porcentaje, y ese sería el modo de obtener beneficios.

También existe la posibilidad de que haya patrocinadores interesados en la web. En ese caso, se podría añadir en algún sitio su marca, o hacer que artistas interesados trabajen en patrocinios con dicha marca, de modo que, al vender sus obras, tanto el artista como la web obtengan beneficios del patrocinio.

En una *webshop* existen muchas maneras de obtener beneficios económicos, por lo que, económicamente, realizar este proyecto es viable.

Viabilidad técnica

Dado que en el apartado del estado del arte se comentan las tecnologías con las que se va a trabajar durante este proyecto, y, además, se ha hecho un estudio de los posibles riesgos y como tomar medidas frente a ellos, en principio no debería haber problemas técnicos.

Ya sea por conocimientos adquiridos anteriormente en base a las tecnologías utilizadas, o por la gestión de riesgos realizada, técnicamente, realizar este proyecto es viable.

Viabilidad legal

Para verificar si este proyecto (*webshop*) es viable legalmente debe cumplir los siguientes requisitos legales:

* Ley 7/1996, de 15 de enero, de Ordenación del Comercio Minorista. [10]
* Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales. [11]
* Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico. [12]
* Ley 3/2014, de 27 de marzo, por la que se modifica el texto refundido de la Ley General para la Defensa de los Consumidores y Usuarios. [13]

Capítulo 4 – Análisis

Capítulo 5 - Diseño

Capítulo 6 - Implementación

Capítulo 7 - Pruebas

Capítulo 8 - 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. |
| [4] | VisualStudio, «Visual Studio Code FAQ,» [En línea]. Available: https://code.visualstudio.com/docs/supporting/FAQ. |
| [5] | Wikipedia, «Angular (framework),» [En línea]. |
| [6] | Capterra, «¿Qué es Visual Paradigm?,» [En línea]. Available: https://www.capterra.ec/software/145716/visual-paradigm. |
| [7] | https://tusalario.es/salario/comparatusalario#/, «Tusalario - Compara tu salario,» [En línea]. |
| [8] | https://www.glassdoor.es/Sueldos/index.htm, «Glassdoor - Burcar sueldos y remuneración,» [En línea]. |
| [9] | Project Management Institute, Guia de los Fundamentos de la Dirección de Proyectos (Guía del PMBok), Project Management Institute, 2009. |
| [10] | Jefatura del Estado, Ley de Ordenación del Comercio Minorista., Agencia Estatal Boletín Oficial del Estado, 1996, 15 de enero. |
| [11] | Jefatura del Estado, Ley de Protección de Datos Personales y garantía de los derechos digitales., Agencia Estatal Boletín Oficial del Estado, 2018, 5 de diciembre. |
| [12] | Jefatura del Estado, Ley de Servicios de la sociedad de la información y de comercio electrónico, Agencia Estatal Boletín Oficial del Estado, 2002, 11 de julio. |
| [13] | Jefatura del Estado, Ley General para la Defensa de los Consumidores y Usuarios, Agencia Estatal Boletín Oficial del Estado, 2014, 27 de marzo. |