**APLICACIÓN WEB PARA LA COTIZACION Y VENTA DE COMPONENTES ELECTRONICOS PARA LA EMPRESA MXT’TECHONOLOGY**

1. **CAPITULO 2**

El presente trabajo abordara las herramientas que se va a ocupar para poder realizar el proyecto de manera eficiente y sin inconvenientes.

Abordaremos sobre lenguajes de código, programas, frameworks y metodologías. Todo esto para acortar y agilizar el trabajo a realizar.

También necesitaremos el uso y conocimiento de un gestor de base de datos como puede ser mysql, MariaDb, MongoDb, SqLite, Oracle.

Estos gestores de bases de datos nos ayudaran a recolectar la información necesaria para el desarrollo de la página, ya que almacenan información en gran cantidad.

1. **HERRAMIENTAS TECNOLOGICAS PARA EL DESARROLLO DE SOFTWARE**

## **BACKEND**

Primeramente, debemos saber qué es y cuál es la participación del backend en la realización de una página web responsiva.

Normalmente los desarrolladores se dividen en 2 partes que son el frontend y backend y suelen trabajar juntos, hay casos excepcionales donde un programador puede dominar ambas partes, pero es mucho trabajo es por eso que se suele dividir este trabajo en más de una persona.

Las herramientas específicas para realizar el backend de esta página web serán lenguajes de código, editores de código y frameworks.

Un desarrollador de este lado (Backend) debe dominar varios lenguajes de programación y frameworks, base de datos.

Backend es aquello que se encuentra del lado del servidor y se encarga de interactuar con base de datos, verificar maniobras de sesiones de usuarios, montar la página en un servidor y servir todas las vistas creadas por el desarrollador frontend.

En este caso el número de tecnologías es mucho menos limitado puesto que la programación backend puede alcanzar lenguajes como PHP, PYTHON, NET, JAVA y otros lenguajes más.

Las bases de datos que se encuentran de este lado son Mysql, MongoDb, Sql y otras.

Php será el principal lenguaje para poder realizar esta página web y darle todas as funciones que vamos a requerir posteriormente, además que este lenguaje de código se caracteriza por estar diseñado realmente para realizar páginas web (Catalunya, 2012).

## **FRONTEND**

El frontend son aquellas tecnologías de desarrollo web del lado del cliente es decir las que corren en el navegador del usuario y que son básicamente tres: HTML, CSS, JavaScript.

Básicamente el frontend se enfoca en el usuario en todas las acciones que pueda realizar para poder interactuar con la página web además de todo lo que puede llegar a ver mientras navega.

Hoy en día se busca tener una buena experiencia de usuario, inmersión y usabilidad, estos son algunos de los objetivos que busca un buen desarrollador frontend usando frameworks y librerías para poder facilitar y cumplir estos trabajos.

Los lenguajes que ocupara el desarrollador de la parte del cliente son los siguientes:

¨¨html, css, java script¨ como principales debe dominar estos lenguajes y también debe usar varios editores de texto como sublime text, visual studio code etc.

Como se mencionaba anteriormente los lenguajes de texto que se ocuparan para la realización de esta página web

**JAVA SCRIPT.–** Es un lenguaje de programación interpretado dialecto del estándar [ECMAScript](https://es.wikipedia.org/wiki/ECMAScript). Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinamico.

Se utiliza principalmente del [lado del cliente](https://es.wikipedia.org/wiki/Cliente_(inform%C3%A1tica)), implementado como parte de un [navegador web](https://es.wikipedia.org/wiki/Navegador_web) permitiendo mejoras en la [interfaz de usuario](https://es.wikipedia.org/wiki/Interfaz_de_usuario) y [páginas web](https://es.wikipedia.org/wiki/P%C3%A1gina_web) dinámicas[3](https://es.wikipedia.org/wiki/JavaScript#cite_note-3)​ y JavaScript del [lado del servidor](https://es.wikipedia.org/wiki/Script_del_lado_del_servidor) (Server-side JavaScript o SSJS). Su uso en [aplicaciones](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_inform%C3%A1tica) externas a la [web](https://es.wikipedia.org/wiki/World_Wide_Web), por ejemplo en documentos [PDF](https://es.wikipedia.org/wiki/PDF), aplicaciones de escritorio (mayoritariamente [widgets](https://es.wikipedia.org/wiki/Widget)) es también significativo.

Desde 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1, una versión de JavaScript. Los navegadores más antiguos soportan por lo menos ECMAScript 3. La sexta edición se liberó en julio de 2015.[4](https://es.wikipedia.org/wiki/JavaScript#cite_note-4)​

JavaScript se diseñó con una sintaxis similar a [C++](https://es.wikipedia.org/wiki/C%2B%2B) y [Java](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)),[5](https://es.wikipedia.org/wiki/JavaScript#cite_note-5)​[6](https://es.wikipedia.org/wiki/JavaScript#cite_note-6)​ aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes. Su relación es puramente comercial, tras la compra del creador de Java (Sun Microsystems) de Nestcape Navigator (creador de LiveScript) y el cambio de nombre del lenguaje de programación.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del [Document Object Model](https://es.wikipedia.org/wiki/Document_Object_Model" \o "Document Object Model) (DOM). Javascript es el único lenguaje de programación que entienden de forma nativa los navegadores.

**CSS. -** Es un lenguaje de [diseño gráfico](https://es.wikipedia.org/wiki/Dise%C3%B1o_gr%C3%A1fico) para definir y crear la presentación de un documento estructurado escrito en un [lenguaje de marcado](https://es.wikipedia.org/wiki/Lenguaje_de_marcado).[2](https://es.wikipedia.org/wiki/CSS#cite_note-2)​ Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en [HTML](https://es.wikipedia.org/wiki/HTML) o [XHTML](https://es.wikipedia.org/wiki/XHTML); el lenguaje puede ser aplicado a cualquier [documento XML](https://es.wikipedia.org/wiki/Extensible_Markup_Language), incluyendo XHTML, [SVG](https://es.wikipedia.org/wiki/Scalable_Vector_Graphics), [XUL](https://es.wikipedia.org/wiki/XML-based_User-interface_Language), [RSS](https://es.wikipedia.org/wiki/RSS), etcétera. Junto con [HTML](https://es.wikipedia.org/wiki/HTML) y [JavaScript](https://es.wikipedia.org/wiki/JavaScript), CSS es una [tecnología](https://es.wikipedia.org/wiki/Tecnolog%C3%ADa) usada por muchos [sitios web](https://es.wikipedia.org/wiki/Sitio_web) para crear páginas visualmente atractivas, [interfaces](https://es.wikipedia.org/wiki/Interfaces) de usuario para [aplicaciones web](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web) y [GUIs](https://es.wikipedia.org/wiki/GUI" \o "GUI) para muchas aplicaciones [móviles](https://es.wikipedia.org/wiki/Smartphone) (como [Firefox](https://es.wikipedia.org/wiki/Firefox) OS).[3](https://es.wikipedia.org/wiki/CSS#cite_note-3)​

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes.[4](https://es.wikipedia.org/wiki/CSS#cite_note-4)​ Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características presentaciones, permitir que varios documentos [HTML](https://es.wikipedia.org/wiki/HTML) compartan un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento.

**HTML. -** Es un lenguaje de marcado que nos permite indicar la estructura de nuestro documento mediante etiquetas. Este lenguaje nos ofrece una gran adaptabilidad, una estructuración lógica y es fácil de interpre­tar tanto por humanos como por máquinas.

Sin embargo, a lo largo de sus diferentes versiones, se han incorporado y suprimido diversas características, con el fin de hacerlo más eficiente y facilitar el desarrollo de páginas web compatibles con distintos navegadores y plataformas (PC de escritorio, portátiles, [teléfonos inteligentes](https://es.wikipedia.org/wiki/Tel%C3%A9fonos_inteligentes), [tabletas](https://es.wikipedia.org/wiki/Tableta_(computadora)), etc.) No obstante, para interpretar correctamente una nueva versión de HTML, los desarrolladores de navegadores web deben incorporar estos cambios y el usuario debe ser capaz de usar la nueva versión del navegador con los cambios incorporados. Normalmente los cambios son aplicados mediante parches de actualización automática ([Firefox](https://es.wikipedia.org/wiki/Firefox), [Chrome](https://es.wikipedia.org/wiki/Google_Chrome)) u ofreciendo una nueva versión del navegador con todos los cambios incorporados, en un sitio web de descarga oficial ([Internet Explorer](https://es.wikipedia.org/wiki/Internet_Explorer)). Por lo que un navegador desactualizado no será capaz de interpretar correctamente una página web escrita en una versión de HTML superior a la que pueda interpretar, lo que obliga muchas veces a los desarrolladores a aplicar técnicas y cambios que permitan corregir problemas de visualización e incluso de interpretación de código HTML. Así mismo, las páginas escritas en una versión anterior de HTML deberían ser actualizadas o reescritas, lo que no siempre se cumple. Es por ello que ciertos navegadores todavía mantienen la capacidad de interpretar páginas web de versiones HTML anteriores. Por estas razones, todavía existen diferencias entre distintos navegadores y versiones al interpretar una misma página web.

CSS será la herramienta principal para diseñar y darle vida a la página, con esto se logrará que la pagina sea intuitiva y atrayente para los clientes.

HTML será de vital importancia ya que es un lenguaje de código que nos permitirá estructurar todo lo que contenga la página web junto a los contenidos que se vayan a agregar.

La idea de esta división es mantener separadas las diferentes partes de un sistema web o software para tener un mejor control, en pocas palabras el objetivo es que el frontend recoja los datos y el backend los procese.

Estas dos capaz que forman una aplicación web son independientes entre sí, es decir que no comparten código, pero intercambian información.

Esta división permite que el acceso a la base de datos solo se haga desde el backend y así el usuario no tenga acceso al código de la aplicación, mientras que la programación del lado del cliente permite que el navegador pueda por ejemplo controlar donde el usuario hace click o accede a sus ficheros.

Con esta división de entornos de usuario de una página web o aplicación web lo que hace es, por ejemplo, iniciar sesión escribiendo su usuario y contraseña en un formulario y a continuación los datos se envían al backend toma esta información que viene desde el html y busca las coincidencias de usuario en la base de datos con una serie de procesos invisibles para el usuario.

En este punto el servido mandaría un mensaje al frontend dándole acceso o no a la aplicación o página web. (Catalunya, 2012)

## **SISTEMA GESTOR DE BASE DE DATOS Y VERSION DE PHP**

Un sistema gestor de base de datos nos va a permitir crear y administrar y almacenar información de la manera más eficiente posible.

Existen 2 tipos de base de datos, los relacionales y no relacionales.

En los gestores de base de datos relacionales tenemos a Mysql, MariaDB, SQlite, PostrgeSql, MicrosoftSqlServer, Oracle.

Gestores de base de datos no relacionales.

MongoDB, Redis, Cassandra, Raven, Neo4j.

Se optará por usar un gestor de base de datos relacional que es el Mysql para poder manejar y administrar de manera eficiente los datos de los usuarios y clientes,

Mysql es el gestor de base de datos más usado por la comunidad y se encuentra en páginas web actuales.

**MySQL.-** Es un [sistema de gestión de bases de datos](https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_bases_de_datos) [relacional](https://es.wikipedia.org/wiki/Modelo_relacional) desarrollado bajo licencia dual: [Licencia pública general](https://es.wikipedia.org/w/index.php?title=Licencia_p%C3%BAblica_general&action=edit&redlink=1)/[Licencia comercial](https://es.wikipedia.org/wiki/Software_propietario) por [Oracle Corporation](https://es.wikipedia.org/wiki/Oracle_Corporation) y está considerada como la [base de datos](https://es.wikipedia.org/wiki/Base_de_datos) de [código abierto](https://es.wikipedia.org/wiki/Open_source) más popular del mundo​ y una de las más populares en general junto a [Oracle](https://es.wikipedia.org/wiki/Oracle_Database) y [Microsoft SQL Server](https://es.wikipedia.org/wiki/Microsoft_SQL_Server), todo para entornos de [desarrollo web](https://es.wikipedia.org/wiki/Desarrollo_web).

Inicialmente desarrollado por [MySQL AB](https://es.wikipedia.org/wiki/MySQL_AB) (empresa fundada por [David Axmark](https://es.wikipedia.org/w/index.php?title=David_Axmark&action=edit&redlink=1), [Allan Larsson](https://es.wikipedia.org/w/index.php?title=Allan_Larsson&action=edit&redlink=1) y [Michael Widenius](https://es.wikipedia.org/wiki/Michael_Widenius)). MySQL AB fue adquirida por [Sun Microsystems](https://es.wikipedia.org/wiki/Sun_Microsystems" \o "Sun Microsystems) en 2008, y ésta a su vez fue comprada por [Oracle Corporation](https://es.wikipedia.org/wiki/Oracle_Corporation) en 2010, la cual ya era dueña desde 2005 de [Innobase Oy](https://es.wikipedia.org/w/index.php?title=Innobase_Oy&action=edit&redlink=1" \o "Innobase Oy (aún no redactado)), empresa [finlandesa](https://es.wikipedia.org/wiki/Finlandia) desarrolladora del motor [InnoDB](https://es.wikipedia.org/wiki/InnoDB" \o "InnoDB) para MySQL.

Al contrario de proyectos como [Apache](https://es.wikipedia.org/wiki/Servidor_HTTP_Apache), donde el software es desarrollado por una comunidad pública y los [derechos de autor](https://es.wikipedia.org/wiki/Derechos_de_autor) del código están en poder del autor individual, MySQL es patrocinado por una [empresa privada](https://es.wikipedia.org/wiki/Empresa_privada), que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de doble licenciamiento anteriormente mencionado. La base de datos se distribuye en varias versiones, una Community, distribuida bajo la Licencia pública general de [GNU](https://es.wikipedia.org/wiki/GNU), versión 2, y varias versiones Enterprise, para aquellas empresas que quieran incorporarlo en productos privativos. Las versiones Enterprise incluyen productos o servicios adicionales tales como herramientas de [monitorización](https://es.wikipedia.org/w/index.php?title=Monitorizaci%C3%B3n_de_sistemas&action=edit&redlink=1) y [asistencia técnica](https://es.wikipedia.org/wiki/Soporte_t%C3%A9cnico) oficial. En [2009](https://es.wikipedia.org/wiki/2009) se creó un [fork](https://es.wikipedia.org/wiki/Bifurcaci%C3%B3n_(desarrollo_de_software)" \o "Bifurcación (desarrollo de software)) denominado [MariaDB](https://es.wikipedia.org/wiki/MariaDB" \o "MariaDB) por algunos desarrolladores (incluido algunos desarrolladores originales de MySQL) descontentos con el modelo de desarrollo y el hecho de que una misma empresa controle a la vez los productos MySQL y [Oracle Database](https://es.wikipedia.org/wiki/Oracle_Database).[3](https://es.wikipedia.org/wiki/MySQL#cite_note-3)​

Está desarrollado en su mayor parte en [ANSI C](https://es.wikipedia.org/wiki/ANSI_C) y [C++](https://es.wikipedia.org/wiki/C%2B%2B).[4](https://es.wikipedia.org/wiki/MySQL#cite_note-4)​ Tradicionalmente se considera uno de los cuatro componentes de la pila de desarrollo [LAMP](https://es.wikipedia.org/wiki/LAMP) y [WAMP](https://es.wikipedia.org/wiki/WAMP).

MySQL es usado por muchos sitios web grandes y populares, como Wikipedia, Google, Facebook,Twitter,Flickr y Youtube.

Además, que es software libre y es de fácil uso y su rendimiento es bueno.

Es fácil de configurar e instalar, el único inconveniente que podemos llegar a tener con este gestor de base de datos es que no trabaja con base de datos de gran tamaño (wikipedia.org, 2022)

**WAMPSERVER. - E**s un acrónimo que significa Windows, Apache, MySQL y PHP. Es un stack o conjunto de soluciones de software que significa que cuando instalas WAMP, estás instalando [Apache](https://www.hostinger.es/tutoriales/que-es-apache/), [MySQL](https://www.hostinger.es/tutoriales/que-es-mysql/) y PHP en tu sistema operativo (Windows en el caso de WAMP). Aunque puede instalarlos por separado, por lo general son empaquetados, y también por una buena razón.

Lo que es bueno saber es que WAMP deriva de [LAMP](https://www.hostinger.es/tutoriales/como-instalar-linux-apache-mysql-php-lamp-en-ubuntu-16-04/) (la L significa Linux). La única diferencia entre estos dos es que WAMP se usa para Windows, mientras que LAMP para sistemas operativos basados ​​en Linux.

Repasemos rápidamente lo que representa cada letra:

* «**W**» significa Windows; también hay LAMP (para Linux) y MAMP (para Mac).
* «**A**» significa Apache. Apache es el software de servidor que se encarga de servir las páginas web. Cuando solicitas ver una página, Apache cumple tu solicitud a través de HTTP y te muestra el sitio.
* «**M**» significa MySQL. El trabajo de MySQL es ser el sistema de gestión de base de datos para tu servidor. Almacena toda la información relevante, como el contenido de tu sitio, los perfiles de usuario, etc.
* «**P**» significa PHP. Es el lenguaje de programación en el cual está escrito WordPress y actúa como aglutinante para toda este stack de soluciones. PHP se ejecuta junto con Apache y se comunica con MySQL.

En lugar de instalar y probar WordPress en tu [cuenta de hosting](https://www.hostinger.es/hosting-web), puedes hacerlo en tu computadora personal (localhost).

WAMP actúa como un servidor virtual en tu computadora (WampServer). Te permite probar todas las funciones de WordPress sin ninguna consecuencia, ya que está ubicado en tu máquina y no está conectado a la web.

En primer lugar, esto significa que no necesitas esperar hasta que los archivos se carguen en tu sitio, y en segundo lugar, esto facilita mucho la creación de copias de seguridad.

Un WampServer acelera el proceso de trabajo tanto para los desarrolladores como para los diseñadores de temas. Además, también obtienes el beneficio de poder jugar con su sitio y tener algo de diversión.

Sin embargo, para que el sitio web funcione, debes adquirir algún tipo de servicio de hosting y dominio. Mira nuestro artículo para principiantes sobre [hosting web](https://www.hostinger.es/tutoriales/que-es-un-hosting) para obtener más información.

## **FRAMEWORKS**

Básicamente es una plantilla que nos ayudara a organizar el código, ya que puede simplificar un proceso y de esa manera aceleramos el trabajo.

El objetivo de un framework literalmente es acelerar el trabajo y reducir el typeo básicamente trabajaremos sobre una plantilla y nos permitirá reutilizar el código una y otra vez.

Las ventajas de usar un framework son reducir el tiempo, evitar errores y facilitar las tareas que vayamos a realizar.

Los frameworks que podemos utilizar son los siguientes:

Angular, Meteor, Django-Ruby On Rails- Laravel- Asp.Net

El que se usará en esta aplicación web será Asp.Net ya que cuenta con servicios de web Php y es ligero y tiene buen rendimiento.

**ASP.NET. -**  Es un framework open source multiplataforma creado por Microsoft. Es decir, estamos ante un entorno de trabajo basado en código abierto que está pensado para el desarrollo y la ejecución de aplicaciones y servicios web modernos. Al ser multiplataforma es posible ejecutar aplicaciones ASP.NET tanto en Windows, Linux y macOS como en contenedores.

La historia de ASP se remonta al año 1996. Por aquel entonces ASP era parte de la versión 3.0 de IIS (Internet Information Services). ASP se creó con el objetivo de poder combinar código HTML con scripts, componentes ActiveX y código en Visual Basic ejecutado desde el servidor. Es a este hecho al que debe su nombre porque ASP significa Active Server Pages o páginas activas desde el servidor.

Respecto a los scripts, recordemos que JavaScript era un lenguaje muy joven, se puede decir que recién creado. JavaScript tenía muchísimo potencial puesto que gracias a él las páginas webs adquirían dinamismo y una “vida” de la que antes carecían por completo.

En el año 97 ya había quejas de los desarrolladores pidiendo mecanismos para escribir código más limpio y poder separar la presentación, el HTML y el JavaScript, del contenido, por aquel entonces ActiveX. Fueron [Scott Guthrie](https://blogs.microsoft.com/blog/author/scottguthrie/) y [Mark Anders](https://www.linkedin.com/in/mark-anders-7972a8149/) los encargados de dirigir este proyecto de cambio que duró alrededor de 4 años y que comenzó llamándose XSP.

El primer enfoque de Mark y Scott con XSP fue utilizar Java. Esto les llevó a querer tener ellos algo análogo a la JVM, la máquina virtual de Java, para no depender de ésta. Empezaron así a trabajar en el CLR, el Common Languaje Runtime, que en esencia era una imitación de la JVM y muchas de sus características como la orientación a objetos o la recolección de basura, algo altamente criticado durante años.

Fue durante el diseño del CLR cuando se decidió crear también un nuevo lenguaje, uno potente, limpio, con una sintaxis similar a la de C y C++ y lo suficientemente versátil como para competir con Java. Este proyecto tomó un nombre en clave, “Proyect Cool”, y tuvo un carácter de alto secreto dentro de Microsoft. A este lenguaje al principio se le llamó Cool, algunos trabajadores lo llamaban “Súper C”. Cool acabó convirtiéndose en C#, lenguaje con el que se re implementó la nueva versión de ASP a la que se le llamó ASP+.

Se intentó que la migración a esta nueva plataforma fuera sencilla y ASP+ se lanzó oficialmente el 11 de Julio del año 2000. En aquella demostración llamó muchísimo la atención la potencia del nuevo framework puesto que era posible combinar ASP con muchos otros lenguajes como COBOL. También se presentaron, entre otras muchas novedades, los nuevos lenguajes de Microsoft, C# y Visual Basic .NET y fue en aquella segunda mitad del año 2000 cuando nació la marca “.NET”.

Tras unos meses de refinamiento y el lanzamiento de varias versiones de evaluación durante el año 2001 ASP.NET fue liberado con gran expectación en enero de 2002 junto con la primera versión del [.NET Framework](https://openwebinars.net/blog/que-es-net-framework/). (Carmona, OpenWebinars, 2022)

## **HERRAMIENTAS ADICIONALES**

Las herramientas adicionales que se utilizara en el desarrollo de esta página web son extensiones o bibliotecas de código abierto para poder agilizar el trabajo y recortar el código.

También se usará páginas de pallets de colores para poder tener un diseño llamativo para el cliente y no perder tiempo en el diseño de la pagina

Esta página acortara proceso del diseño ya que nos brinda un paquete de colores listas para implementarlos en el código

**BOOTSTRAP. -** Es una biblioteca que se usara para poder ordenar la página web ya que nos ofrece variedad de diseños para nuestro sitio web.

Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de [código abierto](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto) para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en [HTML](https://es.wikipedia.org/wiki/HTML) y [CSS](https://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada), así como extensiones de [JavaScript](https://es.wikipedia.org/wiki/JavaScript) adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo [front-end](https://es.wikipedia.org/wiki/Front-end" \o "Front-end).

Bootstrap es el segundo proyecto más destacado en [GitHub](https://es.wikipedia.org/wiki/GitHub)[1](https://es.wikipedia.org/wiki/Bootstrap_(framework)#cite_note-1)​ y es usado por la [NASA](https://es.wikipedia.org/wiki/NASA) y la [MSNBC](https://es.wikipedia.org/wiki/MSNBC), entre otras organizaciones (Wikipedia, 2022)

1. **METODOLOGIA DE DESARROLLO**

Una metodología de desarrollo es básicamente un método para poder realizar nuestro proyecto de manera eficaz.

Ya que con esto se tendrá una guía y una vista clara de cómo debemos realizar nuestro proyecto sin tener inconvenientes y de esa manera podremos adaptar la forma de trabajo a las condiciones del proyecto y su desarrollo a las circunstancias de nuestro entorno.

Se usará una metodología de desarrollo ágil ya que se debe presentar un proyecto que sea eficiente y rápido, las metodologías agiles tienen como característica poner como prioridad al cliente ya que el producto o servicio es para él.

Las ventajas que se obtienen de optar por usar una metodología de desarrollo ágil son varias, el tiempo es una de ellas y también que se trabaja mediante sprints donde podremos presentar avances al cliente y eso es algo importante y fundamental en estas metodologías.

Estas metodologías suelen tener una secuencia de eventos o ciclos de vidas para poder mostrarnos cuál es su forma de trabajo, de esta manera lograremos entender la forma de trabajo y como podremos adaptarlo a nuestro proyecto

**Fig.2.1¨Representación gráfica, ciclo de vida de una metodología ágil. ¨**

**(Fuente:** [**Metodología Agile y sus beneficios | WAM (wearemarketing.com)**](https://www.wearemarketing.com/es/blog/que-es-la-metodologia-agile-y-que-beneficios-tiene-para-tu-empresa.html)**)**

Las características de una metodología ágil son realizar una aplicación o proyecto de manera ¨rápida¨ y de calidad con todos los requisitos que el cliente solicito.

Las desventajas que podemos llegar a tener son en cuanto el alcance, al realizar tantas modificaciones en cuanto a los tiempos y costos del proyecto o producto de manera indirecta el alcance se verá afectado, dependiendo de la metodología a optar se ve la gravedad de las modificaciones al alcance. (Maida, 2018)

La metodología Scrum será la ¨indicada¨ para realizar nuestro proyecto sobre la elaboración de una página web.

El marco “Scrum”, tal como lo conocemos hoy, se introdujo por primera vez en un artículo de la Harvard Business Review en 1986.

[El nuevo juego de desarrollo de nuevos productos (The New Product Development Game)](https://hbr.org/1986/01/the-new-new-product-development-game), escrito por Hirotaka Takeuchi e Ikujiro Nonaka. Takeuchi y Nonaka tomaron el término “Scrum” del rugby, explicando que “como en el rugby, los miembros del equipo se pasan la pelota entre sí, a medida que avanzan como una unidad por el campo de juego”.

Scrum fue luego desarrollado y codificado por Ken Schwaber y Jeff Sutherland en 1995, cuando publicaron su [Manifiesto ágil](https://agilemanifesto.org/) y el [Proceso de Desarrollo SCRUM](http://www.jeffsutherland.org/oopsla/schwapub.pdf).

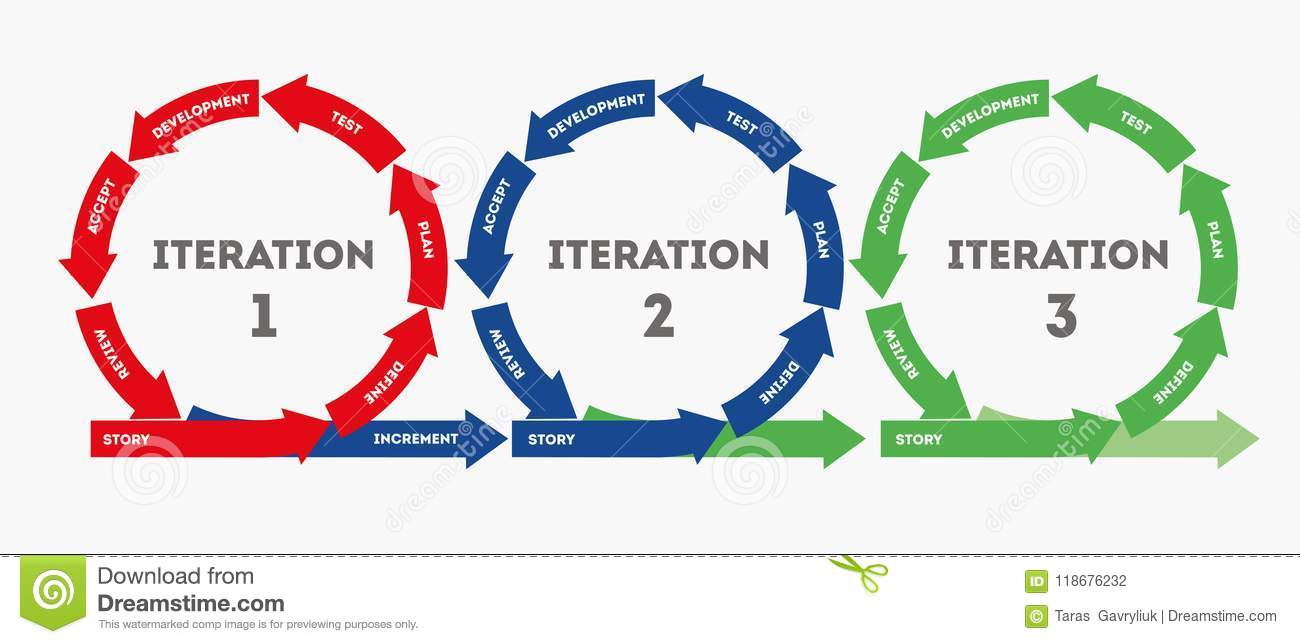
La metodología Scrum de Schwaber y Sutherland fue en parte un rechazo al modelo de cascada de desarrollo de software. En el modelo de cascada, los proyectos se dividen en fases secuenciales, donde los [entregables](https://asana.com/es/resources/what-are-project-deliverables) de cada fase desbloquean la siguiente fase de trabajo. Schwaber y Sutherland creían que los programadores podrían beneficiarse de un enfoque más flexible e iterativo que les permitiera responder y adaptarse continuamente a su entorno para construir el mejor producto final para sus clientes.

Desde su publicación inicial, Schwaber y Sutherland han publicado la [Guía de Scrum](https://www.scrumguides.org/), un documento dinámico que actualizan de forma regular. De acuerdo con la Guía de Scrum, Scrum alienta a “los equipos a observar cuán efectivas son sus técnicas de trabajo y los desafía a evolucionar y mejorarlas continuamente”.

Scrum está diseñado para proyectos que requieren realizar iteraciones rápidas y constantes.

**2.2.1 ¿Qué es una iteración?**

Una iteración es una repetición o reiteración, por lo tanto, un proceso iterativo seria refinar algún avance que se haya realizado en el proyecto.



**Fig. 2.2 ¨Grafico de un ciclo de iteraciones¨**

**Fuente:** [Iteración ilustración del vector. Ilustración de concepto](https://es.dreamstime.com/iteraci%C3%B3n-image118676232)

El objetivo de esta metodología ágil es ayudarnos a realizar un buen trabajo ya que nos va a proporcionar un plan de valores, roles y pautas para poder tener una guía de trabajo, debemos concentrarnos en cada iteración que se realice ya que esta es una característica fundamental de esta metodología.

Como se mencionaba anteriormente se trabajará mediante iteraciones que son los sprints en nuestro proyecto, es aquí donde debemos enfocarnos en cada sprint que generalmente son sesiones de trabajo y generalmente son 2 eventos importantes que están relacionados con el Scrum.

La primera son las reuniones diarias con un tiempo límite de 15 minutos donde se deben coordinar las actividades a realizar.

El segundo evento es el análisis retrospectivo del sprint el objetivo de este evento como su nombre lo indica es analizar el sprint que se realizó para ver si se pueden realizar ajustes y demás cosas.

Como se ve anteriormente esta metodología trabaja mediante eventos que llegaría a ser el ciclo de vida o secuencia de etapas en esta metodología ágil.

Los eventos que usa o requiere esta metodología ágil son seis, los cuales son fundamentales para poder realizar cualquier proyecto junto a esta metodología de desarrollo.

**2.2.2 Secuencia de Eventos**

**Organiza el trabajo pendiente.** Para comenzar un sprint de Scrum, el líder del equipo (también conocido como Scrum Master) identificará qué trabajo extraer de la lista de tareas pendientes, es decir, el trabajo que debe realizarse. Para llevar a cabo el mejor sprint de Scrum posible, se debe de asegurar de que el trabajo pendiente para el producto esté claramente documentado en un solo lugar. Considera usar una [herramienta de gestión de proyectos](https://asana.com/es/uses/project-management) para recopilar toda esta información.

**Realiza una sesión de planificación del sprint.** Antes de que se inicie el sprint de Scrum, se necesita tener un enfoque. Durante la sesión de planificación del sprint, se debe evaluar en qué parte del trabajo pendiente se centrará el equipo durante este sprint de Scrum específico.

**Comienza el sprint de Scrum.** Por lo general, un sprint dura dos semanas, aunque se puede tener sprints más cortos o más largos dependiendo de lo que funcione mejor para el equipo.

Durante el sprint, el equipo trabajará en las tareas pendientes que has establecido durante la sesión de planificación del sprint.

**Organiza reuniones diarias de actualización de Scrum.** Planificar las reuniones con el equipo Scrum durante 15 minutos todos los días. Las reuniones de actualización diarias son tu oportunidad para informar con respecto al trabajo que se está realizando e identificar cualquier obstáculo inesperado que haya surgido.

**Presentar el trabajo durante la revisión del sprint.** Una vez concluida el sprint de Scrum, el equipo debe reunirse para hacer una revisión del sprint. Durante este tiempo, el equipo Scrum presentará el trabajo que está “Terminado” para la aprobación o inspección de los participantes.

**Conversa y reflexiona durante el análisis retrospectivo del sprint.** Al final de cada sprint, se debe tomar un tiempo para analizar cómo se desarrolló y qué podría mejorarse en el futuro. Recuerda que en Scrum se cree en un proceso de mejora continua, así que no temas probar nuevos procesos o reelaborar estrategias que piensas que no son tan efectivas durante el siguiente sprint. (PAMPLONA, 2008)

**Fig. 2.2.2 “Grafico de un Sprint y Orden”**

**Fuente:** [Las 5 ceremonias Scrum: claves para la gestión de procesos (deloitte.com)](https://www2.deloitte.com/es/es/pages/technology/articles/ceremonias-scrum.html)

**2.2.3 Artefactos de Trabajo**

También se debe utilizar artefactos que es una característica de scrum, básicamente es una herramienta que nos permitirá resolver problemas, existen 3 tipos de artefacto: el trabajo pendiente del producto, el trabajo pudiente del sprint y el incremento del producto.

* **El trabajo pendiente del producto.** -Es una lista de maestra del trabajo que se va a realizar, el gerente del proyecto o encargado que es el product owner debe clasificar los elementos de esta lista.

Se debe de tener en cuenta que esta lista no significa que el equipo de trabajo debe realizarlos si no que estos elementos son las opciones de trabajo que pueden tomar los equipos de trabajo durante un sprint.

Los encargados del proyecto deben reordenar y actualizar con frecuencia los trabajos pendientes del producto en función a las modificaciones del cliente mediante requisitos o solicitudes.

* **Trabajo pendiente del sprint. -** Es la serie de trabajos o sprints con los que el equipo se ha comprometido durante el sprint de scrum. Estos elementos se extraen de una lista de trabajo pendiente del producto durante la sesión de planificaciones del sprint y se trasladan al proyecto de planificación del sprint de tu equipo, en caso de haya un equipo claramente.
* **Incremento del producto. -** Esta parte es donde se debe realizar la entrega final de cada sprint. Esta fase puede crear un nuevo producto o función o puede haber mejoras y corrección de errores o cualquier otra variante dependiendo al equipo de trabajo.

En resumen, debemos entregar un incremento siempre y cuando el equipo de trabajo lo apruebe para saber si está terminado o no. (Ailin Orjuela Duarte, 2008)

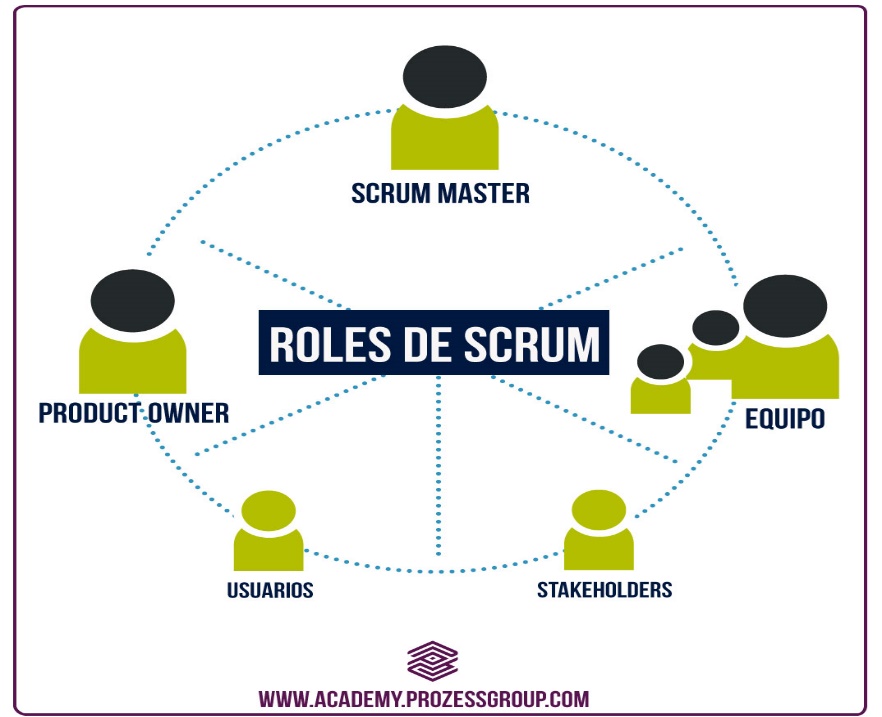
**2.2.4 Roles de Scrum**

Los roles que nos vamos a encontrar en esta metodología son los siguientes:

**Encargado del Producto (Product Owner).** Esta es la persona a cargo de la lista de trabajo pendiente del producto (product backlog). Está conectado a las necesidades del usuario y se centra en transmitir el punto de vista del usuario a su equipo y a otros ejecutivos involucrados. Los buenos encargados de productos aportan claridad sobre qué es lo siguiente que se debe entregar debido a su importancia. En última instancia, deberían ser ellos quienes decidan cuándo algo está listo para ser entregado (con una tendencia a realizar entregas con frecuencia).

**Scrum Master.** El Scrum Master es la persona que dirige los distintos eventos de Scrum. Considéralo el gerente del proyecto y facilitador de Scrum. El Scrum Master debe promover las reuniones diarias de actualización y organizar las reuniones de planificación, revisión y análisis retrospectivo del sprint

**Equipo de Scrum.** El equipo de Scrum es todos los que están trabajando en el sprint. Los miembros del equipo deben auto organizarse y ser colaborativos para lograr el objetivo de Scrum de mejora continua (Ailin Orjuela Duarte, 2008)



**Fig. 2.2.3 “Roles De Scrum”**

**Fuente:** [Roles en un proyecto scrum (prozessgroup.com)](http://www.prozessgroup.com/roles-en-un-proyecto-scrum/)

* + 1. **Principios de la metodología Scrum**
* Existen seis [principios de Scrum](https://www.scrumstudy.com/whyscrum/scrum-principles) que nos ayudaran a aplicar la metodología Scrum y aprovechar los beneficios de Scrum. Son los siguientes:
* **Control.** sobre el proceso empírico. Los equipos Scrum creen en la transparencia, la inspección y la adaptación.
* **Auto organización.** Aunque el equipo Scrum tendrá roles y reglas, cada miembro Scrum está facultado para asumir la responsabilidad de sus tareas y su trabajo. En Scrum se cree que la responsabilidad compartida produce equipos más creativos y dinámicos.
* [**Colaboración.**](https://asana.com/es/resources/team-collaboration-tips) Tu equipo obtendrá los mejores resultados si trabaja en conjunto durante y después del sprint de Scrum.
* **Priorización basada en valores.** El objetivo del sprint de Scrum es ofrecer el mayor valor comercial. Para hacer esto se debe dar prioridad al trabajo desde el inicio del proceso Scrum.
* **Duración limitada (timeboxing).** El proceso Scrum tiene varias actividades basadas en el tiempo, como el sprint en sí, las reuniones diarias de actualización y el análisis retrospectivo. Debido a que Scrum se basa en la creencia de la mejora continua, es importante establecer una duración limitada para el trabajo para pasar a la siguiente tarea y mejorar el trabajo futuro.
* **Desarrollo iterativo.** En Scrum, tu primer producto no será perfecto. Pero al construir de manera iterativa, el equipo podrá adaptarse mejor a las necesidades del cliente y modificar el producto y sus resultados en función de la priorización basada en el valor. (Ailin Orjuela Duarte, 2008)
  + 1. **Valores de la Metodología Scrum**

Para beneficiarse de Scrum, los equipos deben observar los cinco [valores principales de Scrum](https://www.scrumalliance.org/about-scrum/values#:~:text=The%20Scrum%20Guide%20lists%20five,focus%2C%20openness%2C%20and%20respect.), que se definen en la Guía de Scrum:

* **Compromiso:** El equipo Scrum es una unidad, y los miembros del equipo deben confiar entre sí. Los miembros del equipo Scrum están comprometidos con el sprint durante su duración y dedicados a la mejora continua para encontrar la mejor solución.
* **Valor:** Durante un Scrum, el equipo puede encontrar problemas difíciles que no tienen una respuesta exacta. Los equipos Scrum tienen el valor de hacer preguntas abiertas y difíciles y de responder con sinceridad para llegar a la mejor solución.
* **Enfoque:** Durante cualquier sprint de Scrum, el equipo Scrum realizará trabajos extraídos de una lista de tareas pendientes del producto. El equipo Scrum se centra en el trabajo que han elegido de la lista de trabajo pendiente para hacer sus entregables al final de cada sprint.
* **Actitud receptiva:** No todo irá sobre ruedas durante el Scrum. Los miembros del equipo Scrum deben estar abiertos a nuevas ideas y oportunidades que los ayuden a aprender a nivel individual y a mejorar su producto o proceso.
* **Respeto:** La colaboración es la clave de Scrum, y para apoyar la colaboración en equipo, los miembros del equipo deben respetarse entre sí, al Scrum Master y al proceso Scrum.
  + 1. **VENTAJAS DE LA METODOLOGÍA SCRUM**

Scrum es más eficaz para los equipos que necesitan construir y entregar con frecuencia, ya sean “productos” tradicionales como código o nuevas funciones, o “productos” Scrum menos comunes como campañas de marketing o activos creativos.

Los beneficios de Scrum son la agilidad y la flexibilidad. El proceso Scrum puede ayudar a aumentar el trabajo en equipo y alcanzar tus objetivos de manera más efectiva. Además, los equipos Scrum siempre saben exactamente en qué están trabajando, ya que están extrayendo tareas de la lista de trabajo pendiente del producto, y tienen claro cuáles son sus objetivos, ya que todos están en sintonía en relación con lo que significa “Terminado”.

* + 1. **LÍMITES DE SCRUM**
* **Con frecuencia,** los proyectos Scrum pueden sufrir de [corrupción del alcance](https://asana.com/es/resources/what-is-scope-creep) porque el proceso Scrum acepta y fomenta el cambio. Si realizas demasiados cambios o recibes demasiados comentarios discordantes de los clientes, es posible que estés iterando una y otra vez sin resultados reales.
* **Solución:** Asegurarnos de definir con claridad los objetivos del sprint y el incremento de cada sprint. Además, teniendo en cuenta de que todo tu equipo Scrum tenga claro lo que significa “Terminado”, para que no realicen más de lo que hace falta para que el trabajo esté “Terminado”. Si es necesario, implementa un proceso de control de cambios para evitar estos problemas.
* **Los equipos Scrum tienen muchas reuniones;** además de la planificación y revisión de sprints que se programan regularmente, los equipos Scrum también se reúnen a diario para tener una reunión de actualización.
* **Solución:** Si las reuniones diarias de Scrum no te resultan útiles, busca la manera de cambiarlas. Hacer un seguimiento de las reuniones de actualización de un proyecto puede ayudarte a concentrarte solo en los aspectos más útiles.
* **Implementar** Scrum puede ser difícil (aunque no imposible) si no estás en un equipo de desarrollo de software, desarrollo de productos o ingeniería.

**CAPITULO III MARCO APLICATIVO O PROPUESTA DE SOLUCION**

* 1. **INTRODUCCÍON**

Este capítulo abordara sobre la implementación de todas las herramientas y características que tiene la metodología ágil que se seleccionó, requerimientos, historias de usuario, planificación de reuniones y sprints y finalmente el producto backlog

* 1. **REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES**
     1. **REQUERIMIENTOS FUNCIONALES**
* El sistema debe permitir que los clientes busquen un producto por nombre o descripción mediante un filtro de búsqueda.
* Al aprobar un pedido, la solicitud pasará al siguiente paso del flujo de trabajo.
* El sistema permitirá a los usuarios autorizados el ingresar productos y modificarlos para la exposición en la página web.
* La aplicación debe poder utilizarse sin necesidad de instalar ningún software adicional además de un navegador web(Firefox,Chrome,Edge,Opera).
* Los permisos de acceso al sistema podrán ser cambiados por los administradores de la página.
  + 1. **REQUERIMIENTOS NO FUNCIONALES**
* El tiempo para iniciar o reiniciar el sistema no podrá ser mayor a 5 minutos.
* La interfaz de usuario será implementada para navegadores web únicamente con HTML,CSS,JAVASCRIPT y Bootstrap.
* La interfaz del sistema tanto como para el cliente y administrador debe ser intuitiva.
* La aplicación debe poder utilizarse con los navegadores web Chrome, Firefox y Edge.
* El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 4 horas.

**3.3. APLICACIÓN DE LA METODOLOGIA SCRUM**

# **3.3.1 HISTORIAS DE USUARIO**

Con las historias de usuario podremos recolectar y organizar los requerimientos que tenga cada persona (clientes,personal,administradores) y poder priorizar estos pedidos.

|  |  |
| --- | --- |
| **HISTORIAS DE USUARIO** | |
| **Numero:** 1 | **Usuario:** Cliente |
| **Nombre de la Historia:** Interfaz gráfica de la página web | |
| **Prioridad:** | |
| **Programador Responsable:** Pinto Mora Victor Angel | |
| **Descripción:** Como cliente solicito que la interfaz gráfica sea intuitiva para poder realizar y ver los pedidos sin ningún problema y de la manera más rápida. | |
| **Observaciones:** Ninguna | |

|  |  |
| --- | --- |
| **HISTORIAS DE USUARIO** | |
| **Numero:** 2 | **Usuario:** Personal |
| **Nombre de la Historia:** Administración de Categorías | |
| **Prioridad:** | |
| **Programador Responsable:** Pinto Mora Victor Angel | |
| **Descripción:** Como vendedor quiero poder administrar y modificar las categorías para poder organizarlas de la mejor manera y tener el inventario de productos actualizados. | |
| **Observaciones:** Ninguna | |

|  |  |
| --- | --- |
| **HISTORIAS DE USUARIO** | |
| **Numero: 3** | **Usuario:** Cliente |
| **Nombre de la Historia:** Login e ingreso al sistema | |
| **Prioridad:** | |
| **Programador Responsable:** Pinto Mora Victor Angel | |
| **Descripción:** Como cliente solicito que la página web tenga un login para poder realizar acciones con los productos (compras, pedidos, reservas) | |
| **Observaciones:** Ninguna | |

|  |  |
| --- | --- |
| **HISTORIAS DE USUARIO** | |
| **Numero: 4** | **Usuario:** Personal |
| **Nombre de la Historia:** Administración y Modificación de Productos | |
| **Prioridad:** | |
| **Programador Responsable:** Pinto Mora Victor Angel | |
| **Descripción:** Como Personal requiero poder modificar los productos tanto como imágenes, títulos, descripciones o costos de cualquier producto almacenado en la página web. | |
| **Observaciones:** Ninguna | |

|  |  |
| --- | --- |
| **HISTORIAS DE USUARIO** | |
| **Numero: 5** | **Usuario:** Cliente |
| **Nombre de la Historia:** Descripciones Técnicas y Físicas de un Producto | |
| **Prioridad:** | |
| **Programador Responsable:** Pinto Mora Victor Angel | |
| **Descripción:**  Como Cliente solicito que los productos contengan una descripción técnica y específica para poder estar seguro de la compra que vaya a realizar. | |
| **Observaciones:** Ninguna | |

|  |  |
| --- | --- |
| **HISTORIAS DE USUARIO** | |
| **Numero: 6** | **Usuario:** Personal |
| **Nombre de la Historia:** Administración de Pedidos de los Clientes | |
| **Prioridad:** | |
| **Programador Responsable:** Pinto Mora Victor Angel | |
| **Descripción:** Como Personal requiero que los pedidos que se realicen de parte del cliente contengan un número de pedido para poder administrarlos sin problema alguno y ordenar los pedidos de los clientes. | |
| **Observaciones:** Ninguna | |

|  |  |
| --- | --- |
| **HISTORIAS DE USUARIO** | |
| **Numero: 7** | **Usuario:** Cliente |
| **Nombre de la Historia:** AgregarCarrito de compras y pedidos | |
| **Prioridad:** | |
| **Programador Responsable:** Pinto Mora Victor Angel | |
| **Descripción:** Como cliente solicito tener un carrito donde pueda agregar los pedidos que solicite con el objetivo de poder ver la lista de los productos que tengo en dicho carrito. | |
| **Observaciones:** Ninguna | |

|  |  |
| --- | --- |
| **HISTORIAS DE USUARIO** | |
| **Numero: 8** | **Usuario:** Personal |
| **Nombre de la Historia:** Estados de un Pedido | |
| **Prioridad:** | |
| **Programador Responsable:** Pinto Mora Victor Angel | |
| **Descripción:** Como personal solicito que cada pedido contenga un campo donde podamos asignar un estado (Aprobado, Cancelado, Completo, Pendiente) con el objetivo de poder administrarlos de mejor manera y así el cliente tenga conocimiento sobre el estado del producto que solicito. | |
| **Observaciones:** Ninguna | |

# **PRODUCT BACKLOG**

El product backlog (o pila de producto) es un listado de todas las tareas que se pretenden hacer durante el desarrollo de un proyecto.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **N.º Historia**  **Usuario** | **DESCRIPCÍON** | **FECHAS DE INICIO** | **ACEPTADA**  **(SI/NO)** | **OBSERVACIONES** | **GRUPO DE SPRINT** |
| 1 | Realizar interfaz gráfica de la página web | 14/07/2022 | Si | Se realizará una interfaz gráfica lo más intuitiva posible para el cliente. | Primer Sprint |
| 2 | Administración de Categorías | 14/07/2022 | SI |  | Primer Sprint |
| 3 | Login e ingreso al sistema | 14/07/2022 | SI | El Login contara con un inicio de sesión tanto como para clientes y administradores | Primer Sprint |
| 4 | Administración y Modificación de Productos | 04/08/2022 | SI | La modificación que se realice al producto solo será permitida para roles establecidos. | Segundo Sprint |
| 5 | Descripciones Técnicas y Físicas de un Producto | 04/08/2022 | SI |  | Segundo Sprint |
| 6 | Administración de Pedidos de los Clientes | 25/08/2022 | SI |  | Tercer Sprint |
| 7 | AgregarCarrito de compras y pedidos | 25/08/2022 | SI |  | Tercer Sprint |
| 8 | Mostrar los estados de un Pedido |  |  |  | Tercer Sprint |

* + 1. **REUNIONES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Grupo de Sprint** | **FECHAS DE INICIO** | **FECHAS DE**  **CONCLUSÍON** | **FECHA DE REUNIONES** | **OBSERVACIONES**  **TUTOR** |
| Primer Sprint | 28/06/2022 | 14/07/2022 | 14/07/2022 | * Los colores de la página deben ser los que solicito la empresa. * Limitar la agregación de categorías * Debe de existir una opción para recuperar la contraseña tanto como del cliente y del administrador |
| Segundo Sprint | 14/07/2022 | 04/08/2022 | 04/08/2022 |  |
| Segundo Sprint | 14/07/2022 | 04//08/2022 | 04/08/2022 |  |
| Tercer Sprint | 25/08/2022 | 08/09/2022 | 08/09/2022 |  |
| Tercer Sprint | 25/08/2022 | 08/09/2022 | 08/09/2022 |  |
| Cuarto Sprint | 25/08/2022 | 8/09/2022 | 08/09/2022 |  |

* + 1. **ROLES DE USUARIO**

Los roles que nos vamos a encontrar en esta metodología son los siguientes:

* **Encargado del Producto (Product Owner).** Esta es la persona a cargo de la lista de trabajo pendiente del producto (product backlog). Está conectado a las necesidades del usuario y se centra en transmitir el punto de vista del usuario a su equipo y a otros ejecutivos involucrados. Los buenos encargados de productos aportan claridad sobre qué es lo siguiente que se debe entregar debido a su importancia. En última instancia, deberían ser ellos quienes decidan cuándo algo está listo para ser entregado (con una tendencia a realizar entregas con frecuencia). En el presente documento y proyecto el Product Owner es Fernando Mamani dueño actual de la empresa Mxt-Technology.
* **Scrum Master.** El Scrum Master es la persona que dirige los distintos eventos de Scrum. Considéralo el gerente del proyecto y facilitador de Scrum. El Scrum Master debe promover las reuniones diarias de actualización y organizar las reuniones de planificación, revisión y análisis retrospectivo del sprint. En el presente documento y proyecto el Scrum Master es Alejandro Wills Mercado.
* **Equipo de Scrum.** El equipo de Scrum es todos los que están trabajando en el sprint. Los miembros del equipo deben auto organizarse y ser colaborativos para lograr el objetivo de Scrum de mejora continua (Ailin Orjuela Duarte, 2008) El equipo de trabajo será Pinto Mora Victor Angel.
  + 1. **PLANIFICACÍON DE SPRINTS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nº Historia**  **Usuario** | **GRUPO DE SPRINT** | **DESCRIPCÍON** | **FECHAS DE INICIO** | **FECHAS DE**  **CONCLUSION** | **ACEPTADA**  **(SI/NO)** | **OBSERVACIONES** |
| 1 | Primer Sprint | Realizar interfaz gráfica de la página web | 14/07/2022 | 28/07/2022 | Si | Se realizará una interfaz gráfica lo más intuitiva posible para el cliente. |
| 2 | Primer Sprint | Administración de Categorías | 14/07/2022 | 28/07/2022 | SI | La administración solo estará permitida para roles específicos |
| 3 | Primer Sprint | Login e ingreso al sistema | 14/07/2022 | 28/07/2022 | SI | El Login contara con un inicio de sesión tanto como para clientes y administradores |
| 4 | Segundo Sprint | Administración y Modificación de Productos | 04/08/2022 | 25/08/2022 | SI | La modificación que se realice al producto solo será permitida para roles establecidos. |
| 5 | Segundo Sprint | Descripciones Técnicas y Físicas de un Producto | 4/08/2022 | 25/08/2022 | SI |  |
| 6 | Tercer Sprint | Administración de Pedidos de los Clientes | 25/08/2022 | 08/09/2022 | SI |  |
| 7 | Tercer Sprint | AgregarCarrito de compras y pedidos | 25/08/2022 | 08/09/2022 | SI |  |
| 8 | Cuarto Sprint | Mostrar los estados de un Pedido | 25/08/2022 | 8/09/2022 | SI |  |

* + 1. **DISEÑO DE LA BASE DE DATOS**
       1. **DESCRIPCÍON DEL DIAGRAMA A UTILIZAR**

El presente proyecto se utilizará el diagrama entidad relación es un tipo de diagrama de flujo que ilustra cómo las "entidades", como personas, objetos o conceptos, se relacionan entre sí dentro de un sistema.

Los diagramas Entidad Relación se usan a menudo para diseñar o depurar bases de datos relacionales en los campos de ingeniería de software, sistemas de información empresarial, educación e investigación. También conocidos como los ERD o modelos Entidad Relación, emplean un conjunto definido de símbolos, tales como rectángulos, diamantes, óvalos y líneas de conexión para representar la interconexión de entidades, relaciones y sus atributos. Son un reflejo de la estructura gramatical y emplean entidades como sustantivos y relaciones como verbos.

Peter Chen (también conocido como Peter Pin-Shan Chen) actualmente se desempeña como miembro de la facultad de la Universidad Carnegie Mellon ubicada en Pittsburgh y se le atribuye el desarrollo del modelo ER para el diseño de bases de datos en los 70. Mientras trabajaba como profesor adjunto en la Escuela de Administración y Dirección de Empresas Sloan del MIT, publicó un documento influyente en 1976 llamado "Modelo entidad-relación: hacia una visión unificada de los datos".

(Lucidchart, 2022)

En un sentido más amplio, la representación de la interconexión de las cosas se remonta hasta, al menos, la Antigua Grecia, con los trabajos de Aristóteles, Sócrates y Platón. Se ha visto más recientemente en las obras del siglo XX y XIX de filósofos y lógicos, como Charles Sanders Peirce y Gottlob Frege.

En la década del 60 y 70, Charles Bachman (arriba) y A.P.G. Brown trabajaron con los primeros antecesores del enfoque de Chen. Bachman desarrolló un tipo de diagrama de estructura de datos que lleva su nombre: "el diagrama de Bachman". Brown publicó escritos sobre el modelado de los sistemas del mundo real. James Martin agregó mejoras al ERD. El trabajo de Chen, Bachman, Brown, Martin y otros también contribuyó al desarrollo del lenguaje unificado de modelado (UML), ampliamente utilizado en el diseño de software.

Los componentes principales para poder realizar un diagrama entidad relación son las entidades las relaciones y los atributos.

Las entidades pueden ser (personas, lugares, eventos, ventas, clientes, departamentos, estudiantes, productos y otros). Una entidad se muestra en una tabla de base de datos en un sistema de datos y cada fila representa una instancia de la entidad.

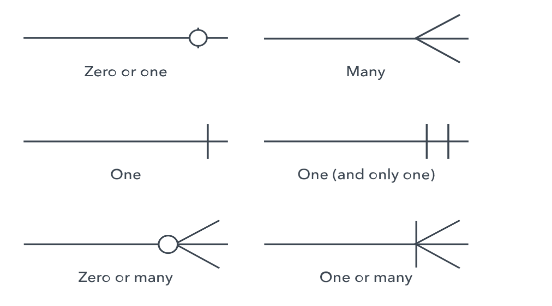
Los atributos son la información detallada que se obtiene para las entidades. En un sistema de una pagina web los atributos de los clientes pueden ser nombres, direcciones y numero de contado y demás.

Las relaciones nos describen como van a interactuar las entidades entre sí. En las relaciones tenemos varios tipos de relación(Uno a muchos, Uno a uno, Muchos a Muchos).

**Los beneficios de usar este diagrama entidad relación son los siguientes:**

* Nos permitirá modelar la base de datos sin atender a un modelo en especial
* Permite visualizar la base de datos desde un alto nivel de abstracción.
  + - 1. **SIMBOLOGÍA DEL DIAGRAMA ENTIDAD RELACÍON**

Normalmente los símbolos que se suelen ocupar para poder realizar un diagrama entidad relación son los siguientes:

****

**ENTIDAD ESTILOS DE CARDINALIDAD**

**Tipo de entidad:** un grupo de cosas que se pueden definir, como estudiantes o atletas, mientras que la entidad sería el estudiante o atleta específico. Otros ejemplos son clientes, autos o productos.

**Conjunto de entidades:** es igual que un tipo de entidad, pero se define en un momento determinado, como por ejemplo estudiantes que se inscribieron en una clase el primer día. Otros ejemplos son clientes que realizaron una compra en el último mes o autos registrados actualmente en Florida. Un término relacionado es una instancia, en la que una persona determinada o un auto específico podría ser una instancia del conjunto de entidades.

**Categorías de entidades:** las entidades se clasifican en fuertes, débiles o asociativas. Una entidad fuerte se puede definir únicamente por sus propios atributos, en cambio, una entidad débil no. Una entidad asociativa es aquella que relaciona entidades (o elementos) dentro de un conjunto de entidades.

**Claves de entidad:** se refiere a un atributo que únicamente define una entidad en un conjunto de entidades. Las claves de entidad se dividen en superclave, clave candidata o clave primaria. **Superclave:**un conjunto de atributos (uno o más) que juntos definen una entidad en un conjunto de entidades. **Clave candidata:**es una superclave mínima, es decir, contiene el menor número posible de atributos para seguir siendo una superclave. Un conjunto de entidades puede tener más de una clave candidata. **Clave primaria:**es una clave candidata seleccionada por el diseñador de la base de datos para identificar únicamente al conjunto de entidades. **Clave extranjera:**identifica la relación entre las entidades.

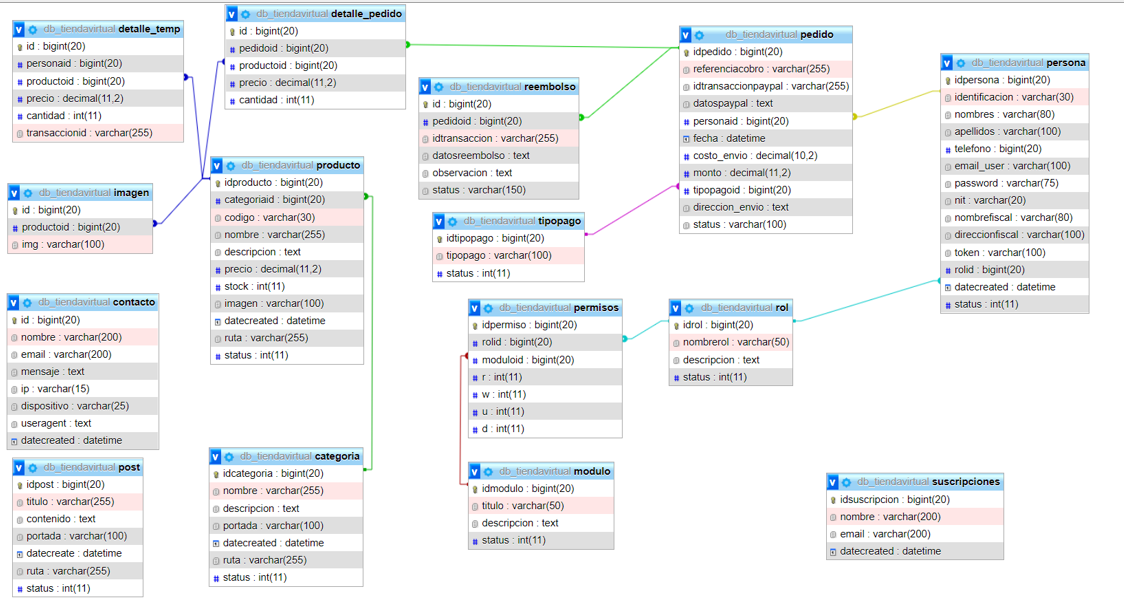
**Relación recursiva:**la misma entidad participa más de una vez en la relación.

El atributo de una propiedad o característica de una entidad. A menudo se muestra como un óvalo o círculo. Atributo descriptivo una propiedad o característica de una relación (frente a una entidad).

La categoría de los atributos se clasifica en simples, compuestos y derivados, así como de valor único o de valores múltiples. Simples: significa que el valor del atributo es mínimo y ya no puede dividirse, como un número de teléfono. Compuestos: los suba tributos surgen de un atributo. Derivados: los atributos se calculan o derivan de otro atributo, por ejemplo, la edad se calcula a partir de la fecha de nacimiento. Los **valores múltiples** se denotan más de un valor del atributo, como varios números de teléfono para una persona.

La cardinalidad define los atributos numéricos de la relación entre dos entidades o conjuntos de entidades. Las tres relaciones cardinales principales son uno a uno, uno a muchos y muchos a muchos. Un **ejemplo de uno a uno** sería un estudiante asociado a una dirección de correo electrónico. Un **ejemplo de uno a muchos (o muchos a uno, en función de la dirección de la relación) sería** un estudiante que se inscribe en muchos cursos, y todos esos cursos se asocian a ese estudiante en particular. **Un ejemplo de muchos a muchos sería**los estudiantes en grupo están asociados a múltiples miembros de la facultad y a su vez los miembros de la facultad están asociados a múltiples estudiantes. La cardinalidad puede estar del lado opuesto o del mismo, en función de dónde se muestran los símbolos. Las restricciones de cardinalidad son los números máximos o mínimos que se aplican a una relación.

* + - 1. **DIAGRAMA E.R DEL PROYECTO**

****

* + - 1. **DESCRIPCÍON DE CADA UNA DE LAS CLASES DEL DIAGRAMA**

**Entidad: Relación Usuarios**

La tabla usuarios almacenara todos los datos de los clientes de la página web de mxt-technology en relación a el nombre, apellidos, correo electrónico, contraseñas y su respectivo rol.

**Entidad: Relación Pedidos**

La tabla pedidos almacenara el id del producto junto al id del usuario que solicito dicho producto, de que departamento o localidad, costo del producto, estado, fecha y hora.

**Entidad: Relación Categorías**

La tabla categorías contendrá solo un id y nombre de categoría para poder dividirlas y ordenarlas de manera más efectiva.

**Entidad: Relación Productos**

La tabla productos contendrá y almacenará el id, categoría a la que pertenecerá, nombre, descripción, precio, stock (disponibilidad), oferta, fecha e imagen de producto.

**Entidad: Relación Líneas y Pedido**

La tabla de líneas y pedido es una subclase de Pedidos y Productos donde almacenaremos el id de esta clase, id pedido, id producto y las unidades.

* + - 1. **DICCIONARIO DE DATOS**

|  |  |  |
| --- | --- | --- |
| **TABLA : USUARIOS** | | |
| **Atributo** | **Tipo Dato** | **Descripción** |
| Id | Int | Registrará a los usuarios con un identificador para organizarlos mejor |
| Nombre | String | Almacenara el nombre de usuario de los clientes o administradores registrados |
| Apellidos | String | Almacenara los apellidos de los clientes |
| Email | String | Almacenara el correo de los usuarios |
| Password | String | Almacenaremos la contraseña de los usuarios registrados |
| Rol | String | Campo para almacenar el rol de cada usuario del sistema |
| Imagen | Img | Campo para ingresar la imagen del usuario |

|  |  |  |
| --- | --- | --- |
| **TABLA : PEDIDOS** | | |
| **Atributo** | **Tipo Dato** | **Descripción** |
| Id | Int | Identificador para registrar el pedido |
| Usuario\_id | Int | Identificador del usuario y pedido |
| Departamento | String | Campo para almacenar el departamento del usuario cliente |
| Localidad | String | Campo para poder ingresar la localidad del cliente |
| Dirección | String | Campo para almacenar la dirección del cliente |
| Coste | Int | Registramos el costo del pedido total |
| Estado | String | Campo para ingresar el estado de un pedido |
| Fecha | Date | Campo para registrar fechas de los pedidos realizados |
| Hora | Time | Campo para registrar la hora del pedido |

|  |  |  |
| --- | --- | --- |
| **TABLA : LINEAS\_PEDIDOS** | | |
| **Atributo** | **Tipo Dato** | **Descripción** |
| Id | Int | Identificador de líneas y pedido |
| Pedido\_id | Int | Identificador del id del pedido |
| Producto\_Id | Int | Identificador del producto |
| Unidad | Int | Campo para ingresar la cantidad de pedidos |

|  |  |  |
| --- | --- | --- |
| **TABLA : PRODUCTOS** | | |
| **Atributo** | **Tipo Dato** | **Descripción** |
| Id | Int | Identificador para registrar el producto |
| Categoria\_id | Int | Campo para ingresar el id de la categoría a la que pertenezca el producto registrado |
| Nombre | String | Nombre del producto registrado |
| Descripción | String | Campo para ingresar datos o detalles sobre el producto registrado |
| Precio | Int | Campo para almacenar el costo que tendrá el producto registrado |
| Stock | Int | Con este campo podremos consultar la disponibilidad de algún producto |
| Oferta | String | Campo que nos permitirá saber si el producto pertenece o tiene alguna oferta disponible |
| Fecha | Date | Campo para registrar fechas de los productos |
| Imagen | Img | Campo para almacenar y agregar una imagen al producto |

|  |  |  |
| --- | --- | --- |
| **TABLA : CATEGORIAS** | | |
| **Atributo** | **Tipo Dato** | **Descripción** |
| Id | Int | Identificador para registrar o listar las categorías |
| Nombre | String | Campo para asignar un nombre a la categoría que se vaya a crear |

**CAPITULO IV FASE DE CONSTRUCCION**

# **4.1 HERRAMIENTAS DE IMPLEMENTACION**

Abarcaremos todas las herramientas que utilizaremos con el objetivo de la elaboración de la pagina web tanto en el backend como en el frontend, también veremos los patrones de diseño que usaremos para tener el código de manera ordenada

# **4.2 HERRAMIENTAS DE DISEÑO**

# **4.2.1 PATRONES DE DISEÑO**

Los **patrones de diseño** son soluciones habituales a problemas que ocurren con frecuencia en el diseño de software. Son como planos prefabricados que se pueden personalizar para resolver un problema de diseño recurrente en tu código.

No se puede elegir un patrón y copiarlo en el programa como si se tratara de funciones o bibliotecas ya preparadas. El patrón no es una porción específica de código, sino un concepto general para resolver un problema particular. Puedes seguir los detalles del patrón e implementar una solución que encaje con las realidades de tu propio programa.

A menudo los patrones se confunden con algoritmos porque ambos conceptos describen soluciones típicas a problemas conocidos. Mientras que un algoritmo siempre define un grupo claro de acciones para lograr un objetivo, un patrón es una descripción de más alto nivel de una solución. El código del mismo patrón aplicado a dos programas distintos puede ser diferente.

Una analogía de un algoritmo sería una receta de cocina: ambos cuentan con pasos claros para alcanzar una meta. Por su parte, un patrón es más similar a un plano, ya que puedes observar cómo son su resultado y sus funciones, pero el orden exacto de la implementación depende de ti.

# **4.2.2 VENTAJAS DE LOS PATRONES DE DISEÑO**

Los patrones de diseño son un juego de herramientas de **soluciones comprobadas a** problemas habituales en el diseño de software. Incluso aunque nunca te encuentres con estos problemas, conocer los patrones sigue siendo de utilidad, porque te enseña a resolver todo tipo de problemas utilizando principios del diseño orientado a objetos.

Los patrones de diseño definen un lenguaje común que puedes utilizar con tus compañeros de equipo para comunicaros de forma más eficiente. Podrías decir: “Oh, utiliza un singleton para eso”, y todos entenderían la idea de tu sugerencia. No habría necesidad de explicar qué es un singleton si conocen el patrón y su nombre.

# **4.2.3 PATRON DE DISEÑO MVC**

MVC (Modelo-Vista-Controlador) es un patrón en el diseño de software comúnmente utilizado para implementar interfaces de usuario, datos y lógica de control. Enfatiza una separación entre la lógica de negocios y su visualización. Esta "separación de preocupaciones" proporciona una mejor división del trabajo y una mejora de mantenimiento.

El modelo define qué datos debe contener la aplicación. Si el estado de estos datos cambia, el modelo generalmente notificará a la vista (para que la pantalla pueda cambiar según sea necesario) y, a veces, el controlador (si se necesita una lógica diferente para controlar la vista actualizada).

Volviendo a nuestra aplicación de lista de compras, el modelo especificará qué datos deben contener los artículos de la lista (artículo, precio, etc.) y qué artículos de la lista ya están presentes.

La vista define cómo se deben mostrar los datos de la aplicación.

En nuestra aplicación de lista de compras, la vista definiría cómo se presenta la lista al usuario y recibiría los datos para mostrar desde el modelo.

El controlador contiene una lógica que actualiza el modelo y / o vista en respuesta a las entradas de los usuarios de la aplicación.

Entonces, por ejemplo, nuestra lista de compras podría tener formularios de entrada y botones que nos permitan agregar o eliminar artículos. Estas acciones requieren que se actualice el modelo, por lo que la entrada se envía al controlador, que luego manipula el modelo según corresponda, que luego envía datos actualizados a la vista.

Sin embargo, es posible que también se desee actualizar la vista para mostrar los datos en un formato diferente, por ejemplo, cambiar el orden de los artículos de menor a mayor precio o en orden alfabético. En este caso, el controlador podría manejar esto directamente sin necesidad de actualizar el modelo.

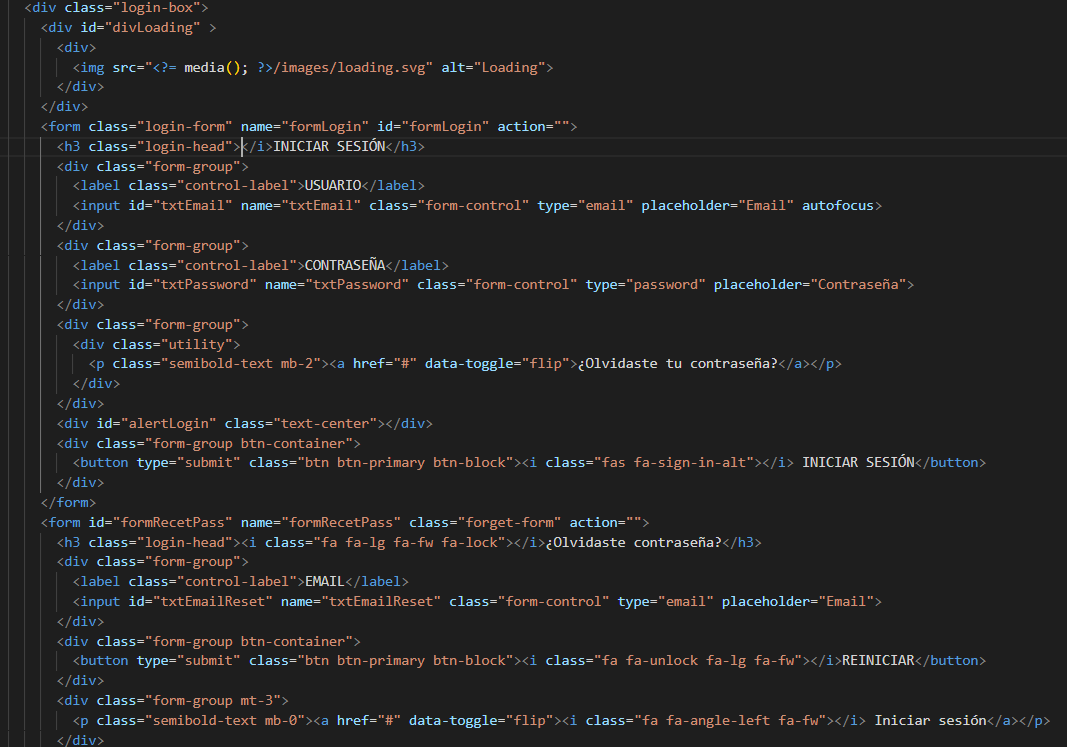
# **ENTORNO DE DESARROLLO**

**4.2.1 IMPLEMENTACION DE ACCESO AL SISTEMA E INTERFAZ GRAFICA**

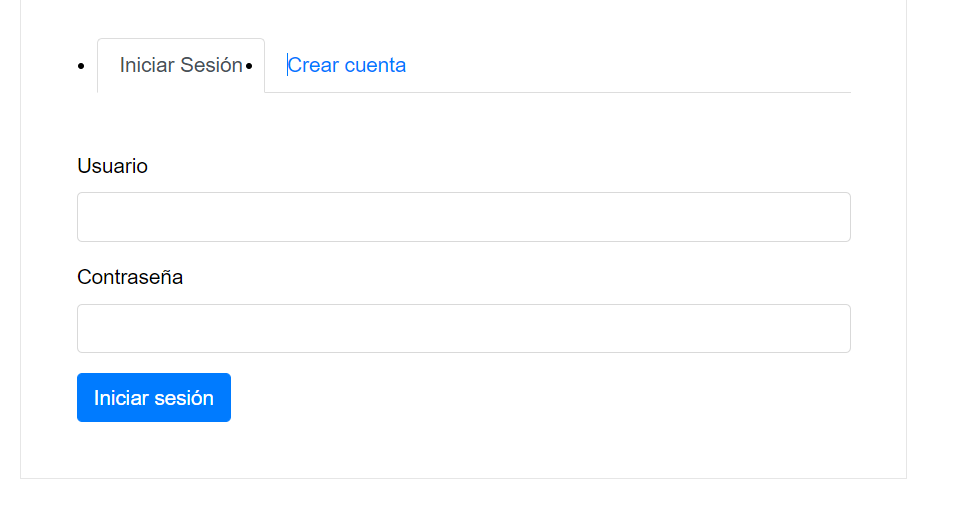
En el primer sprint se llevó a cabo la realización de la interfaz gráfica y login de la página con el objetivo de que sea intuitiva para el cliente y que pueda acceder con un rol a la página tanto como cliente o como administrador.



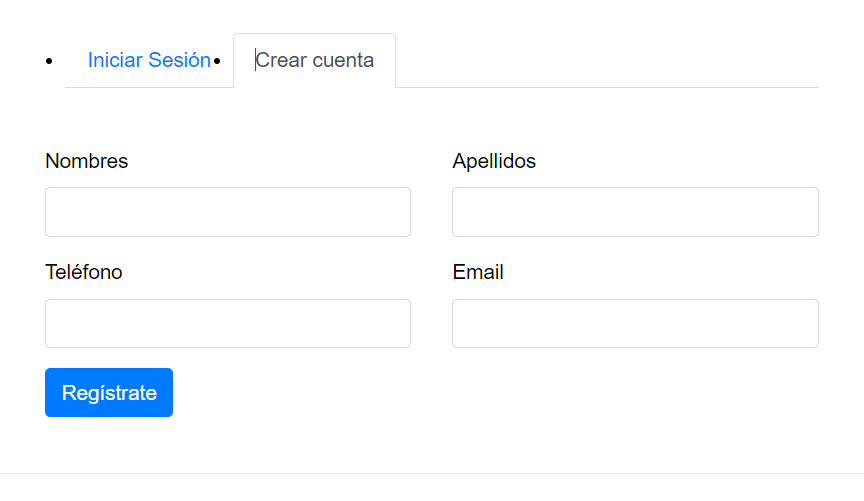
**Figura 4.1.1 Registro**

****

**Figura 4.1.2 Login**

****

**Figura 4.1.3 Login Interfaz Grafica**

****

**Figura 4.1.4 Registro Interfaz Grafica**

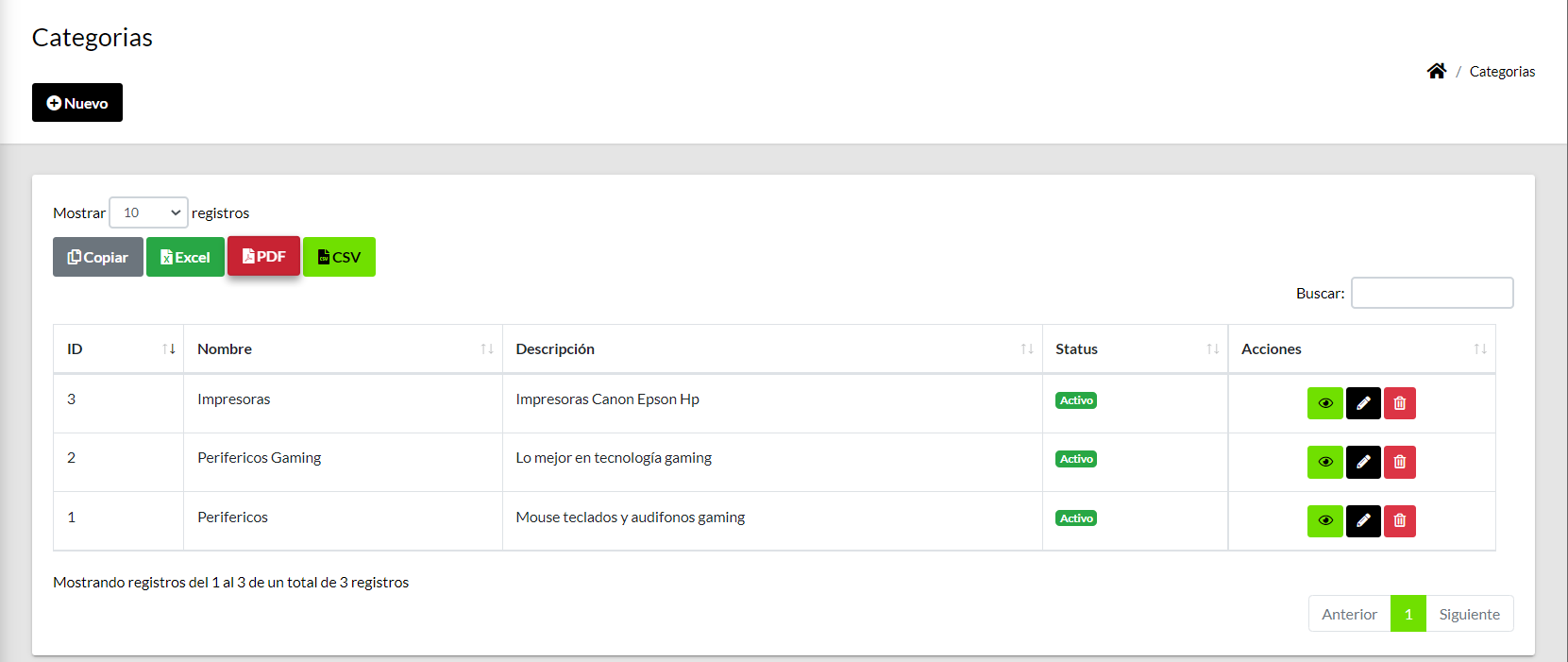
****

**Figura 4.1 Login Interfaz grafica**

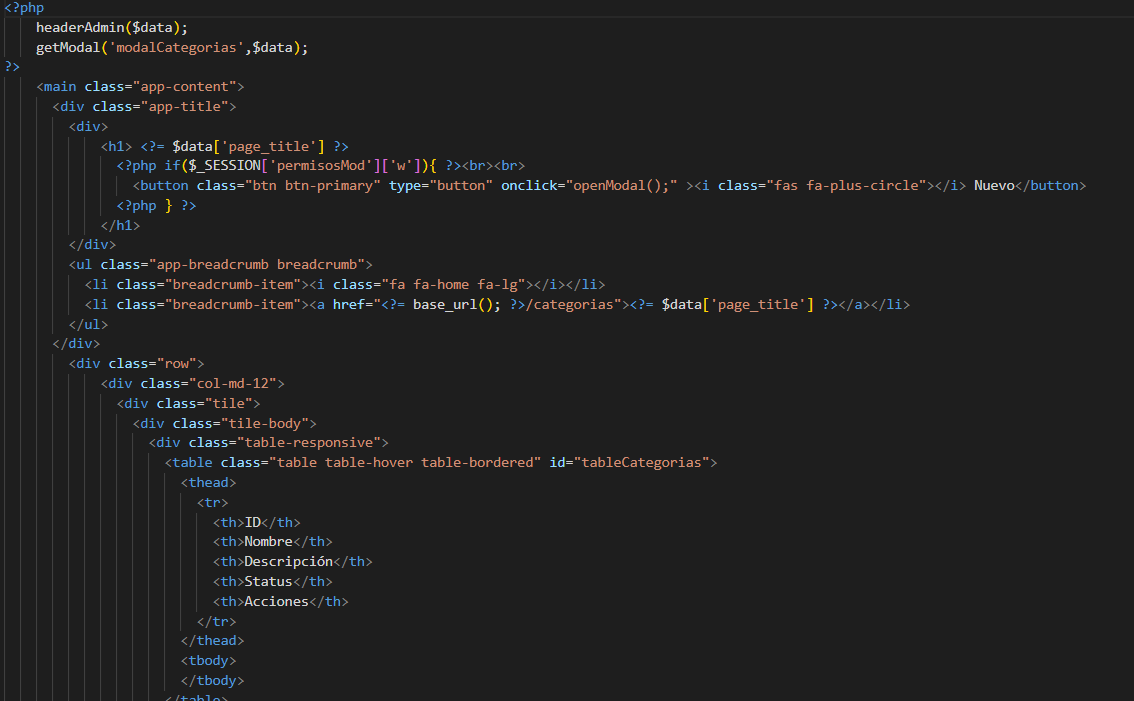
* + 1. **IMPLEMENTACION DE CATEGORIAS DE PRODUCTOS**

Se implemento la lista de categorías a la página web, el usuario al crear una categoría esta se mostrará en la página de inicio de la página.

**Figura 4.1.3 Gestión de Categorías**



**Figura 4.1.4 Gestión de Categorías**

****

**Figura 4.1.4 Gestión de Categorías**

# **4.4 DIAGRAMA DE ACTIVIDADES DE ADMINISTRACION DE USUARIOS**

**SUPERVISOR**

Este rol tendrá todas las funciones disponibles, la función esencial de este rol es la modificación de los usuarios registrados en el sistema.

**CLIENTE**

Este rol tendrá como función principal la realización de pedidos sobre los productos.

**VENDEDOR**

La función primordial de este rol será la venta de los productos ya que podrá modificar los pedidos que realicen los clientes

# **FASE DE PRUEBAS**

En esta sección veremos los requerimientos funcionando, que se solicitó tanto de parte del cliente como de los usuarios, también se verificara los campos de validación de la pagina web y por ultimo la generación de reportes.

# **4.5.1 REQUERIMIENTOS FUNCIONALES IMPLEMENTADOS**

* El sistema permite que los clientes tengan un filtro de búsqueda para los productos.
* Al aprobar un pedido del cliente pasa al siguiente paso de flujo de trabajo.
* El sistema permite que solo usuarios autorizados como (supervisor) realice modificaciones sobre los productos.
* La página web se puede utilizar en los navegadores como (Edge,Chrome,Firefox,Opera)
* Los permisos de la pagina pueden ser modificados por el único rol (supervisor) de la página.

# **4.5.2 CAMPOS DE VALIDACION**

Una regla de validación es una forma de limitar la información que se escribe en un campo de tabla o en un control (como un cuadro de texto) de un formulario. Con el texto de validación se puede incluir un mensaje que sirva de ayuda a los usuarios que escriben datos no válidos.

Estos métodos de validación de datos se pueden usar de forma aislada o combinados entre sí. Los tipos de datos no son opcionales y reportan la validación de tipo de datos más básica.

A continuación, se realiza la muestra de un ejemplo sobre un campo de validación de correo electrónico.

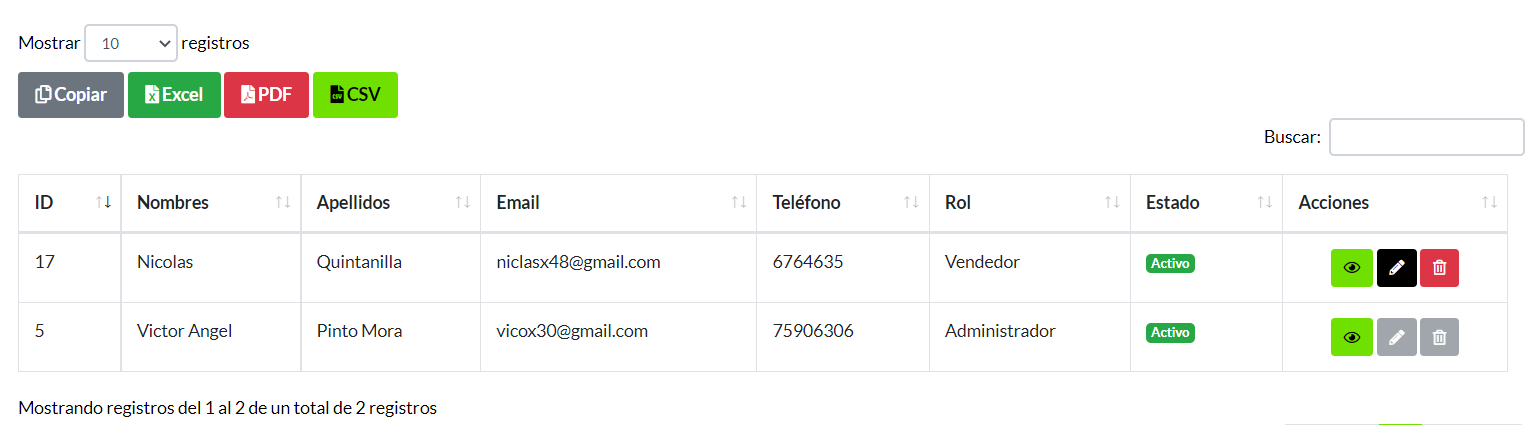


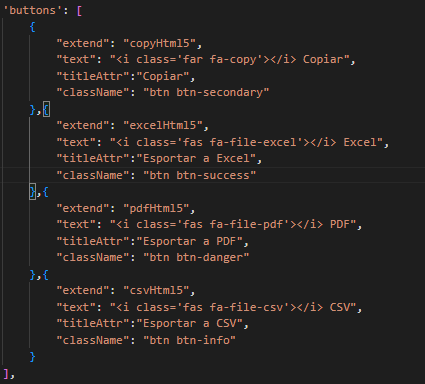


**Figura 4.1.6 Validación de campo correo electrónico**

# **4.5.3 GENERACION DE REPORTES**

La generación de reportes se realiza mediante funciones de JavaScript generando formatos como (Excel,pdf,csv). Con el objetivo de facilitar trabajo a los usuarios de la pagina en la sección de administración.

 **Figura 4.1.7 Generación de reportes**



**Figura 4.1.8 Generación de reportes código JavaScript**

# **4.6 REQUERIMIENTOS DEL SISTEMA**

En el mundo del desarrollo de software y aplicaciones, los requerimientos son críticos. Son los que definen la funcionalidad y el propósito de una pieza particular de software o aplicación. Sin estos bien definidos, es difícil crear algo que satisfaga las necesidades y expectativas del usuario.

En esta sección veremos todas las necesidades técnicas en cuanto a hardware y software de un dispositivo móvil o un computador, todo esto con el fin de poder ingresar a la pagina web.

# **4.6.1 REQUERIMIENTO DEL SOFTWARE**

Los requerimientos de software son simplemente una descripción de lo que un programa de software en particular debe hacer. Actúan como pautas para que los desarrolladores creen un producto funcional que satisfaga las necesidades de los usuarios.

En requerimientos de software solo necesitaremos un sistema operativo donde podamos instalar un navegador para poder ingresar a la página web elaborada.

# **4.6.2 REQUERIMIENTOS DEL HARDWARE**

Para la elaboración de páginas web se necesita contar con los elementos básicos, tanto en la parte de hardware como de software, por ello, en lo referente al hardware se recomienda tener una computadora con los mínimos requisitos que hoy en día se utilizan, aunque algunos piensan que con un simple bloc de notas en la plataforma de windows es suficiente y lo puedes tener en una computadora con muy bajas características de hardware, sin embargo, en la práctica al estar diseñando y creando páginas web, empleamos diversas herramientas de software, muy en específico en el ambiente de windows se requiere de la plataforma CS6 (dreamweaver, flash y fireworks) además de usar el explorador de windows y el internet explorer al mismo tiempo con las herramientas para el diseño, razón por la cual, la computadora que se emplea en el diseño debe tener características de hardware bastante robustas para hacer del diseño un placer y no una tortura, esperando unos minutos para que una aplicación abra y poder trabajar en ella, y tener que cerrar las demás para abrir una nueva, y esperar que la computadora no se bloquee durante algún proceso complejo como es manejar sonido o video.

Los requerimientos para el uso de la pagina web en cuanto a hardware será mínima, ya que simplemente se debe de contar con un navegador web como (Firefox,Opera,Edge,Chrome) para poder utilizar todas las funciones que tenga tanto como para el cliente y como administrador.

# **4.7 INSTALACION Y CONFIGURACION DEL SISTEMA**

# **4.7.1 DOMINIO**

El “dominio” es el nombre que identifica un sitio web. Por regla general, un dominio está formado por tres partes: las famosas “www” que hacen referencia a la red mundial de internet (World Wide Web), seguidas por el nombre de la organización, y finalmente por el tipo de organización (com,es,net,org,coop,etc).

Sin embargo, la red se basa en direcciones IP y no en nombres de dominio. Estas direcciones IP son simples números (p.ej. 185.34.20.10) que, obviamente son difíciles de recordar a la hora de introducir una dirección web en el navegador; es por eso, que es mucho más fácil introducir el nombre del dominio ubicado en un servidor específico que actúa como una agenda telefónica en primera instancia (servidor primario). En caso que este servidor primario fallara, el sistema se va a valer de un servidor secundario que responderá por el primario.

# **4.7.2 HOSTING**

El espacio disponible: tendrás que contratar una cantidad de MB o GB suficiente para todos lo que vayas a meter en la web. Lo más práctico aquí es comprar sólo el que realmente necesites: a veces hay empresas que ofrecen espacio ilimitado, lo cual es muy tentador… pero que no te va a hacer falta. ¿Para qué pagar por una mansión de veinte estancias por planta, cuando con un piso de cuatro habitaciones vas que chutas?

El ancho de banda: o el tráfico mensual que podrá soportar. Cada vez que un usuario entre en tu web, en su ordenador o dispositivo se le descargarán tus fuentes, imágenes, textos y funciones de la misma. Estos datos hay que enviarlos desde el servidor donde los tienes almacenados, hasta el usuario. Y eso pesa una serie de MB y GB. Si los usuarios que recibes cada mes sobrepasan ese volumen de datos, la web dejará de estar operativa hasta el siguiente mes. Por lo tanto, aquí siempre es práctico pasarse un poquito, que sobre un poco de ancho de banda: primero porque no puedes permitirte que tu web se cierre unos días siquiera, y segundo porque ese “excedente” es el que se consumirá conforme tu web vaya creciendo y ganando tráfico.

La base de datos: Esto quizá no era tan importante hace años, pero hoy en día es ya una necesidad. La base de datos es clave para alojar los elementos y proyectos web si vas a utilizar lenguajes más complejos que el HTML base, o un CMS (lo vemos en el siguiente punto). La mayoría de webs modernas funcionan así, ya que facilita mucho cualquier trabajo de diseño, mantenimiento o actualización de la misma.

Hay otros elementos a tener en cuenta, por ejemplo, la cantidad de cuentas de correo que te incluya el hosting que elijas, las cuentas FTP (para subir y bajar ficheros de tu web), pero los tres primeros son sin duda los factores más importantes.

# **4.7.3 WEBSITE**

Ahora que ya tienes un dominio, y un lugar donde alejar tu sitio web, lo único que necesitas es ponerte con ello (o encargárselo al profesional que haya de hacerlo). La página web se construye a partir de un lenguaje de programación como (Html,javascript,php,asp,jsp,Python,ruby,etc).

# **4.7.3 UTILIZACION DEL SISTEMA**

**Bibliografía**

Ailin Orjuela Duarte, M. R. (2008). Las metodologias de Desarrollo Agil como una oprtunidad para la ingenieria del software educativo. En G. d. CICOM, *Las Metodologias de Desarrollo Agil como una Oportunidad para la Ingenieria del Software Educativo* (pág. 165.166.167). Colombia: Universidad de Pamplona.

Carmona, J. (18 de Marzo de 2022). *OpenWebinars*. Obtenido de OpenWebinars: https://openwebinars.net/blog/que-es-aspnet-y-cuales-son-sus-puntos-fuertes/

Carmona, J. (s.f.). *OpenWebinars*.

Carmona, J. (s.f.). *OpenWebinars*.

Catalunya, F. P. (2012). Introduccion al Frontend y Backend. En A. F. Mestres, *Introduccion al Frontend y Backend* (págs. 6,7). Catalunya: Universidad Oberta De Catalunya.

Lucidchart. (Junio de 2022). *Lucidchart.com*. Obtenido de lucidchart: https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion

Maida, E. G. (2018). Tesis en licenciatura en Sistemas y Computacion. En M. Esteban, *Metodologias de Desarrollo De Software* (págs. 3,4,9,10). Argentina: Universidad Catolica de Argentina.

PAMPLONA, G. D. (2008). LAS METODOLOGIAS DEL DESARROLLO AGIL . En M. R. AILIN ORJUELA DUARTE, *LAS METODOLOGIAS DEL DESARROLLO AGIL COMO UNA OPORTUNIDAD PARA LA INGENIERIA DEL SOFTWARE* (págs. 166-167). MEDELLIN.

Wikipedia. (14 de Mayo de 2022). *Wikipedia.org*. Obtenido de Wikipedia.org: https://es.wikipedia.org/wiki/Bootstrap\_(framework)#Origen

wikipedia.org. (30 de Abril de 2022). *Wikipedia*. Obtenido de Wikipedia: https://es.wikipedia.org/wiki/MySQL