

**PROYECTO DE GRADO**

“DESARROLLO DE UNA APLICACIÓN WEB PARA LA COTIZACION Y VENTA DE COMPONENTES ELECTRONICOS PARA LA EMPRESA MXT-TECHNOLOGY**”**

**POSTULANTE:** PINTO MORA VICTOR ANGEL

**TUTOR:** GRADO Y NOMBRE COMPLETO DEL TUTOR

**Proyecto de Grado para optar al Grado Académico de Técnico**

**Superior en Sistemas Informáticos**

**COCHABAMBA- BOLIVIA**

**GESTIÓN 2022**

# **RESUMEN**

**TITULO: “**APLICACIÓN WEB PARA LA COTIZACION Y VENTA DE COMPONENTES Y ACCESORIOS DE LA EMPRESA MXT-TECHNOLOGY**”**

**AUTOR (ES):** PINTO MORA VICTOR ANGEL

# **PROBLEMATICA**

Actualmente la empresa cuenta con dos sucursales y un sistema de venta y todas las ventas son realizadas de manera común y tradicional, si el cliente realiza la búsqueda de algún producto deberá dirigirse a alguna de las 2 sucursales para poder ver si el producto está disponible y así ver las características de tal producto incluido su precio, también cuenta con números de referencia donde se puede realizar las mismas acciones, pero son pocos los que lo realizan ya que requiere tiempo de ambas partes y son pocos los clientes los que realizan estas funciones mediante llamadas.

La empresa Mxt-technology cuenta con redes sociales (Facebook y WhatsApp) donde se atiende de manera ¨virtual¨ pero no están actualizadas tanto como en productos, diseños o atención, por lo tanto, pasa lo mismo que en las atenciones a través de llamadas son pocos los casos

# **OBJETIVO GENERAL**

Realizar una página web para que el cliente pueda acceder fácilmente a la visualización de los productos y también a las características de dicho producto con el objetivo de incrementar ventas y facilitar al cliente los distintos métodos que hay hoy en día para poder realizar una compra o simplemente ver que productos o servicios me ofrece dicha empresa. Y así lograr incrementar el movimiento de los productos que llegaría a ser el aumento de ventas.

# **CONTENIDO**

El sistema por realizar será una página web con base de datos y registro para usuarios (clientes) que quieran adquirir un producto de la tienda, también se puede ver los productos sin estar registrado, el registro será requisito solo al realizar una acción como la compra. La función del sistema básicamente será poder visualizar e interactuar con los productos que sean publicados en la página web de la empresa Mxt-Techonology. Poder realizar pedidos o simplemente ver que productos están disponibles. La metodología que se va a aplicar a este proyecto va a ser una metodología incremental ya que se va a requerir realizar un gran avance en ¨ un corto tiempo¨ y también se requerirá aporte del cliente y de esa manera lograr interactuar con el cliente para poder aumentar funcionalidades y correcciones a la página.

**DEDICATORIA**

**Este proyecto de grado está dedicado a:**

Mi familia que siempre me apoyo en todo este proceso de estudio, especialmente a mi madre que con su paciencia y esfuerzo me ha permitido lograr este objetivo tan importante en mi vida.

Al señor Fernando Mamani y sra Teresa Cutili Quispe quienes fueron los que me dieron una gran oportunidad de trabajo donde pude obtener varias habilidades que posteriormente pude aplicarlas en mis estudios.

También resaltar el apoyo emocional y económico que me brindaron para poder realizar mi proyecto sin ningún problema alguno.

A mis hermanos que con su cariño y apoyo incondicional me dieron la seguridad y confianza que necesite para la elaboración de este proyecto.

**AGRADECIMIENTOS**

Quiero agradecer a mi familia y especialmente a mi madre por el apoyo incondicional.

A mis compañeros y amigos que hicieron de estos años los mejores en cuanto a mis estudios.

A los docentes de mi carrera por que han aportado su granito de arena en mi formación, dedicándonos todo el tiempo, junto a su paciencia y dedicación.

**INDICE**

[**RESUMEN** 2](#_Toc116312682)

[**PROBLEMATICA** 2](#_Toc116312683)

[**OBJETIVO GENERAL** 2](#_Toc116312684)

[**CONTENIDO** 3](#_Toc116312685)

[**CAPÍTULO I – MARCO REFERENCIAL** 12](#_Toc116312686)

[**1 INTRODUCCIÓN** 12](#_Toc116312687)

[**1.2 ANTECEDENTES** 12](#_Toc116312688)

[**1.3 PLANTEAMIENTO Y FORMULACION DEL PROBLEMA** 13](#_Toc116312689)

[**1.3.2 IDENTIFICACION DEL PROBLEMA** 14](#_Toc116312690)

[**1.3.3 FORMULACION DEL PROBLEMA** 15](#_Toc116312691)

[**1.4** **OBJETIVOS** 15](#_Toc116312692)

[**1.4.1 OBJETIVO GENERAL** 15](#_Toc116312693)

[**1.4.2 OBJETIVOS ESPECIFICOS** 15](#_Toc116312694)

[**1.5 JUSTIFICACION** 16](#_Toc116312695)

[**1.5.1 JUSTIFICACION SOCIAL** 16](#_Toc116312696)

[**1.5.2 JUSTIFICACION TECNICA** 16](#_Toc116312697)

[**1.5.3** **JUSTIFICACION ECONOMICA** 17](#_Toc116312698)

[**1.6** **ALCANCES** 17](#_Toc116312699)

[**1.6.1 ALCANCES DEL PROYECTO** 17](#_Toc116312700)

[**1.6.2 ALCANCE DEL PRODUCTO** 17](#_Toc116312701)

[**1.7 LIMITES** 18](#_Toc116312702)

[**1.8 ENFOQUE METODOLÓGICO** 18](#_Toc116312703)

[**1.8.1 METODOLOGIA DE DESARROLLO** 18](#_Toc116312704)

[**1.8.1.1 METODOLOGIA SCRUM** 19](#_Toc116312705)

[**1.8.2 TECNICAS DE RECOLECCION DE INFORMACION** 20](#_Toc116312706)

[**1.8.3 POBLACION DEL OBJETO DE ESTUDIO** 21](#_Toc116312707)

[**CAPÍTULO II – MARCO TEORICO** 22](#_Toc116312708)

[**2.1 INTRODUCCIÓN** 22](#_Toc116312709)

[**2.2 SISTEMA DE INFORMACIÓN** 22](#_Toc116312710)

[**2.3 HERRAMIENTAS PARA EL DESARROLLO DE SOFTWARE** 22](#_Toc116312711)

[**2.3.1 CSS** 22](#_Toc116312712)

[**2.3.2 HTML** 23](#_Toc116312713)

[**2.4 HERRAMIENTAS PARA LA IMPLEMENTACION DEL BACKEND** 25](#_Toc116312714)

[**2.4.1 PHP** 26](#_Toc116312715)

[**2.4.2 SGBD** 27](#_Toc116312716)

[**2.4.3 MYSQL** 27](#_Toc116312717)

[**2.4.4 OTRAS HERRAMIENTAS ADICIONALES** 28](#_Toc116312718)

[**2.4.5 FRAMEWORKS** 29](#_Toc116312719)

[**2.4.6 BOOTSTRAP** 29](#_Toc116312720)

[**2.4.7 PATRON DE DISEÑO MVC** 30](#_Toc116312721)

[**2.4.8 VISUAL STUDIO CODE** 32](#_Toc116312722)

[**2.4.9 PHP MAILER** 32](#_Toc116312723)

[**2.4.10 HOSTING Y DOMINIO** 32](#_Toc116312724)

[**2.5 METODOLOGIAS ÁGILES** 33](#_Toc116312725)

[**2.5.1 DEFINICION DE METODOLOGIAS AGILES** 33](#_Toc116312726)

[**2.5.2 PRINCIPIOS DEL MANIFIESTO AGIL** 34](#_Toc116312727)

[**2.5.3 CARACTERISTICAS DE LAS METODOLOGIAS AGILES** 36](#_Toc116312728)

[**2.5.4 METODOLOGIAS AGILES VS TRADICIONALES** 37](#_Toc116312729)

[**2.6 METODOLOGIA SCRUM** 39](#_Toc116312730)

[**2.6.1 HISTORIA** 39](#_Toc116312731)

[**2.6.2 FORMA DE TRABAJO** 40](#_Toc116312732)

[**2.6.3 CICLO DE VIDA** 40](#_Toc116312733)

[**2.6.4 ROLES DE USUARIO** 42](#_Toc116312734)

[**2.6.5 CARACTERISTICAS** 43](#_Toc116312735)

[**2.6.6 VENTAJAS** 44](#_Toc116312736)

[**2.6.7 DESVENTAJAS** 44](#_Toc116312737)

[**2.7 FUNDAMENTOS TEORICOS DEL SISTEMA** 45](#_Toc116312738)

[**CAPITULO III-MARCO APLICATIVO O PROPUESTA DE SOLUCION** 45](#_Toc116312739)

[**3.1 INTRODUCCION** 45](#_Toc116312740)

[**3.2 REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES** 46](#_Toc116312741)

[**3.2.1 REQUERIMIENTOS FUNCIONALES** 46](#_Toc116312742)

[**3.2.2 REQUERIMIENTOS NO FUNCIONALES** 46](#_Toc116312743)

[**3.3 APLICACIÓN DE LA METODOLOGIA SCRUM** 47](#_Toc116312744)

[**3.3.1 HISTORIAS DE USUARIO** 47](#_Toc116312745)

[**3.3.2 PRODUCT BACKLOG** 52](#_Toc116312746)

[**3.3.4 REUNIONES** 54](#_Toc116312747)

[**3.3.5 ROLES DE USUARIO** 55](#_Toc116312748)

[**3.3.6 PLANIFICACION DE SPRINTS** 57](#_Toc116312749)

[**3.4 DISEÑO DE LA BASE DE DATOS** 59](#_Toc116312750)

[**3.4.1 DESCRIPCION DEL DIAGRAMA A UTILIZAR** 59](#_Toc116312751)

[**3.4.2 SIMBOLOGIA Y ELEMENTOS** 61](#_Toc116312752)

[**3.4.3 DIAGRAMA ENTIDAD RELACION DEL PROYECTO** 63](#_Toc116312753)

[**3.4.4 DESCRIPCION DE CADA UNA DE LAS TABLAS DE LA B.D** 63](#_Toc116312754)

[**3.4.5 DICCIONARIO DE DATOS** 64](#_Toc116312755)

[**3.5 FASE DE CONSTRUCCION** 65](#_Toc116312756)

[**3.5.1 HERRAMIENTA DE IMPLEMENTACION** 65](#_Toc116312757)

[**3.5.2 HERRAMIENTAS DE DISEÑO** 65](#_Toc116312758)

[**3.5.2.1 PATRONES DE DISEÑO** 65](#_Toc116312759)

[**3.5.2.2 VENTAJAS DE LOS PATRONES DE DISEÑO** 66](#_Toc116312760)

[**3.5.2.3 PATRON DE DISEÑO MVC** 67](#_Toc116312761)

[**3.5.3 ENTORNO DE DESARROLLO** 68](#_Toc116312762)

[**3.5.3.1 IMPLEMENTACION DEL ACCESO AL SISTEMA** 68](#_Toc116312763)

[**3.5.3.2 IMPLEMENTACION DE LA INTERFAZ GRAFICA** 71](#_Toc116312764)

[**3.5.3.3 IMPLEMENTACION DE ADMINSITRACION DE PRODUCTOS** 72](#_Toc116312765)

[**3.5.3.4 IMPLEMENTACION DE DESCRIPCIONES TECNICAS PARA LOS PRODUCTOS** 73](#_Toc116312766)

[**3.5.3.5 IMPLEMENTACION PARA LA ADMINISTRACION DE PEDIDOS DE LOS CLIENTES** 74](#_Toc116312767)

[**3.5.3.6 IMPLEMENTACION DE CARRITO DE COMPRAS** 75](#_Toc116312768)

[**3.5.3.7 IMPLEMENTACION DE LOS ESTADOS DE LOS PEDIDOS** 76](#_Toc116312771)

[**3.5.3.8 IMPLEMENTACION DE MODIFICACION DE CLIENTES** 77](#_Toc116312772)

[**3.5.3.9 IMPLEMENTACION DE CATEGORIAS** 78](#_Toc116312773)

[**3.5.4. IMPLEMENTACION DE LISTA DE PEDIDOS** 79](#_Toc116312774)

[**3.5.4.1 IMPLEMENTACION DE REPORTES** 80](#_Toc116312775)

[**3.5.4 DIAGRAMA DE ACTIVIDADES ADMINISTRACION DE USUARIOS** 81](#_Toc116312776)

[**3.5.5 FASE DE PRUEBAS** 81](#_Toc116312777)

[**3.5.6 REQUERIMIENTOS DEL SISTEMA** 81](#_Toc116312778)

[**3.5.6.1 REQUERIMIENTOS DE HARDWARE** 82](#_Toc116312779)

[**3.5.6.2 REQUERIMIENTOS DEL SOFTWARE** 83](#_Toc116312780)

[**3.5.7 INSTALACION Y CONFIGURACION DEL SISTEMA** 83](#_Toc116312781)

[**3.5.7.1 DOMINIO** 83](#_Toc116312782)

[**3.5.7.2 HOSTING** 83](#_Toc116312783)

[**3.5.7.3 WEBSITE** 84](#_Toc116312784)

[**3.6 CONCLUSIONES Y RECOMENDACIONES** 85](#_Toc116312785)

[**3.6.1 CONCLUSIONES** 85](#_Toc116312786)

[**3.6.2 RECOMENDACIONES** 85](#_Toc116312787)

[**BIBLIOGRAFIA** 85](#_Toc116312788)

**ILUSTRACIONES**

# **CAPÍTULO I – MARCO REFERENCIAL**

**TITULO DEL PROYECTO**

# **1 INTRODUCCIÓN**

El desarrollo de las páginas web es muy necesario en estos tiempos debido a que ayudan en mucho en cuanto a tener impacto en la sociedad, específicamente en el ambiente tecnológico y puede ser utilizado con varios objetivos con el fin de aprovechar los beneficios que estos traen. El proyecto busca elaborar una página web para realizar las ventas de manera totalmente virtual sin la necesidad de un personal de la empresa, exceptuando la confirmación de pago y así lograr incrementar el movimiento de productos de todas las sucursales con las que cuente la empresa. El fin del presente documento es definir busca acordar todos los límites y objetivos del proyecto y el tiempo y la inversión

# **1.2 ANTECEDENTES**

La empresa Mxt-techonology cuenta con 2 sucursales la primera ubicada en Esteban Arce y la otra en la Blanco Galindo y av., Perú. El propietario o fundador de esta empresa es Teresa Cutili Quispe junto a su socio Fernando Mamani. La empresa fue fundada en el mes de noviembre del 2010. La misión de esta empresa es poder brindar productos de tecnología con un asesoramiento correspondiente a cada producto y así facilitar las compras que requiera el cliente. La empresa Mxt tiene la visión de poder mantenerse firme frente a las nuevas adaptaciones tecnológicas y poder seguir brindando los productos informáticos y también sus servicios durante muchos años.

La empresa brinda productos informáticos para todo ambiente laboral informático, cuenta con equipos de cómputo y accesorios o dispositivos de red, cuenta con periféricos para varios equipos y también con portátiles y varios productos de tecnología

# **1.3 PLANTEAMIENTO Y FORMULACION DEL PROBLEMA**

**1.3.1 Descripción del Problema**

Actualmente la empresa cuenta con dos sucursales y un sistema de venta y todas las ventas son realizadas de manera común y tradicional, si el cliente quiere realizar la búsqueda de algún producto deberá dirigirse personalmente a alguna de las 2 sucursales para poder ver si el producto está disponible y así ver las características de tal producto incluido su precio, también cuenta con números de referencia donde se puede realizar las mismas acciones, pero son pocos los que lo realizan ya que requiere tiempo de ambas partes y son pocos los clientes los que realizan estas funciones mediante llamadas.

La empresa Mxt-Technology cuenta con redes sociales (Facebook y WhatsApp) donde se debería de atender y realizar publicaciones de los productos, pero es mínimo el movimiento de gente en ambas redes sociales ya que no hay ningún encargado de las redes sociales y las publicaciones de productos son mínimas. Por lo tanto, el problema se origina en el tiempo que le toma a alguien el estar realizando publicaciones o contestando a clientes sobre informaciones de los productos.

Al realizar estas observaciones o diagnósticos del funcionamiento actual de la empresa se identificaron los siguientes problemas:

**a)** Actualmente la empresa no tiene una alternativa mejor que la principal que es “la venta física en cualquiera de las dos sucursales”.

**b)** Hay poca actividad en redes sociales y no cuentan con una página web propia de la empresa.

**c)** La atención por llamada es muy mínima y morosa debido al tiempo que requiere.

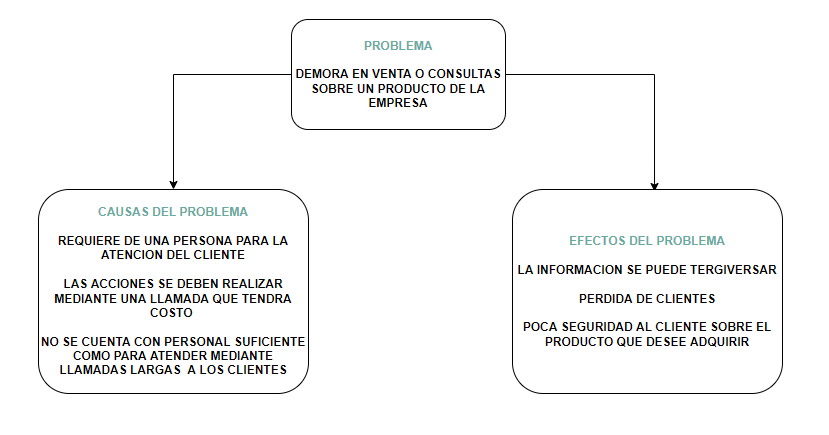
**d)** Los clientes en redes sociales son pocos y es un problema grande debido a que es una tienda de informática o productos tecnológicos.

**e)** Hay mucha pérdida de tiempo en manejar Facebook y WhatsApp como redes sociales para la venta de productos ya que se requiere de una persona que este contestando y mandando fotos y características sobre algún producto que desee el cliente.

# **1.3.2 IDENTIFICACION DEL PROBLEMA**

Como podemos observar en los antecedentes la empresa solo realiza la exposición, ventas y cotizaciones mediante las sucursales de manera física, el cliente debe aproximarse a alguna de estas sucursales para ver si hay la disponibilidad de algún producto o si tan solo desea averiguar el costo del producto, otra opción que opta por usar la empresa es la del teléfono, ya que realiza cotizaciones a través de ellas o pedidos.

Estos procedimientos requieren de una persona que este al tanto del teléfono celular o teléfono de la empresa, provocando de manera indirecta la pérdida de tiempo de ambas partes, además del gasto de las llamadas.



**Ilustración 1: Esquema del Problema**

# **1.3.3 FORMULACION DEL PROBLEMA**

¿De qué manera podemos mejorar el proceso de ventas y cotizaciones sobre los productos de la empresa MXT-TECHNOLOGY?

# **OBJETIVOS**

En esta sección veremos los objetivos que se van a requerir para la elaboración del proyecto

# **1.4.1 OBJETIVO GENERAL**

Realizar una página web para que el cliente pueda acceder fácilmente a la visualización de los productos y también a las características de dicho producto con el objetivo de incrementar ventas y facilitar la compra de un producto mediante un pedido que solicitara el cliente o simplemente ver que productos o servicios me ofrece dicha empresa. Y así lograr incrementar el movimiento de los productos que llegaría a ser el aumento de ventas. ´

# **1.4.2 OBJETIVOS ESPECIFICOS**

* Realizar los talleres de utilización de la página web tanto como para el cliente y el personal del trabajo de la empresa.
* Que el cliente logre visualizar a detalle los productos expuestos en la página web.
* Utilizar un patrón de diseño para la elaboración de la página web con el objetivo de mantener organizado el código.
* Cada producto debe tener la mayor información posible para llamar la atención del cliente y de esa manera asegurar su compra del producto.
* La página debe de contar con sistema de login tanto como para el cliente y administrador o personal de la empresa
* Manejar roles en la página que permitirán dividir las funciones del personal del trabajo y cliente
* Elaborar un manual sobre el funcionamiento del sistema

# **1.5 JUSTIFICACION**

# **1.5.1 JUSTIFICACION SOCIAL**

La elaboración de este sistema de ventas busca tener una relevancia social ya que se busca normalizar las compras mediante internet y agilizar este proceso. Se busca facilitar la compra y exposición de productos tecnológicos favoreciendo tanto al cliente como al vendedor y así tener un alcance más grande en cuanto a las ventas.

# **1.5.2 JUSTIFICACION TECNICA**

La implementación de la página web permitirá lograr un mayor control y exposición sobre los productos, beneficiando tanto como al cliente como al vendedor. El beneficio será en gran parte para el cliente ya que solo tendrá que realizar el pedido del producto que desee mediante a página web.

Además de la exposición de productos se podrá saber cuáles son los productos o categorías con más movimientos, estos productos estarán en la página principal de la página.

# **JUSTIFICACION ECONOMICA**

La implementación de la página web para la empresa mxt-technology permitirá tener mayor alcance sobre la exposición y control de sus productos y de la misma manera podrá reducir gastos en publicidad digital en redes sociales.

Ya que normalmente estas empresas que manejan productos informáticos electrónicos suelen optar por contratar diseñadores gráficos y personal para realizar publicidad y manejo de redes sociales todo esto con el fin de aumentar la publicidad de sus productos mediante la publicidad.

# **ALCANCES**

# **1.6.1 ALCANCES DEL PROYECTO**

* Elaborar un manual operativo del sistema
* Realizar test de validación de usuario
* Tener conocimiento sobre DBMS
* Aplicar un patrón de diseño para la elaboración de la página web
* Diseñar un sistema intuitivo de la pagina

# **1.6.2 ALCANCE DEL PRODUCTO**

* Desarrollar una medida de validación de usuario para el sistema.
* Registrar productos en la página.
* Registrar usuarios para posteriormente almacenarlos en la base de datos.
* Exponer productos ingresados en la base de datos.
* Tener un módulo de autenticación para todos los roles.

# **1.7 LIMITES**

* Solo un determinado rol podrá realizar modificaciones a la página.
* El diseño será a pedido del administrador de la empresa, con un tema en específico con los colores y logo de la empresa.
* El sistema contara con 4 roles esenciales que son (Administrador, Supervisor, Vendedor, Cliente).

# **1.8 ENFOQUE METODOLÓGICO**

A continuación, se presentará el enfoque metodológico que se va a requerir para este proyecto.

# **1.8.1 METODOLOGIA DE DESARROLLO**

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

Ya que con esto tendremos 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.

Usaremos 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 podemos obtener 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.

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.

# **1.8.1.1 METODOLOGIA SCRUM**

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.

Las iteraciones son repeticiones o reiteraciones, por lo tanto, un proceso iterativo seria refinar algún avance que se haya realizado en el proyecto.

# **1.8.2 TECNICAS DE RECOLECCION DE INFORMACION**

Hay muchas maneras de recolectar información en una investigación. El método elegido por el investigador depende de la pregunta de investigación que se formule. Algunos métodos de recolección de información incluyen encuestas, entrevistas, pruebas, evaluaciones fisiológicas, observaciones, revisión de registros existentes y muestras biológicas.

La recolección de información se realizó mediante entrevistas y observaciones.

Una entrevista es una interacción que involucra al investigador y a un(os) participante(s) en que las preguntas se formulan en persona, por teléfono o incluso de manera electrónica (correo electrónico o Internet). Durante una entrevista, se hacen preguntas para obtener información detallada sobre el participante acerca del tema de estudio. Las preguntas pueden ser similares a las formuladas en una encuesta.

Las observaciones son registros tomados que no requieren participación. Estos registros se hacen mientras los participantes están involucrados en conductas rutinarias y se utilizan como un indicador de lo que los participantes hacen, en lugar de apoyarse completamente en los relatos de participantes sobre su propia conducta. Un ejemplo sería un investigador observando los planes educativos usados en un aula por un maestro de escuela pública.

En cuanto a las entrevistas se tuvo charlas junto al propietario de la empresa donde se aplicará este sistema web, juntando así gran parte de la información necesaria para poder realizar este proyecto de manera eficiente.

Las observaciones fueron otra gran herramienta que ayudo a recolectar información que no se conocía y que posteriormente serian de gran ayuda

# **1.8.3 POBLACION DEL OBJETO DE ESTUDIO**

La población objetivo, en otras palabras, se conforma por el conjunto de individuos sobre los que se efectúa una evaluación o [análisis estadístico](https://economipedia.com/definiciones/estudio-estadistico.html).

Para determinar la población objetivo se pueden hacer preguntas como ¿Quién se está viendo afectado o está causando el fenómeno de interés? ¿Cuáles son las características de estas personas y dónde se ubican geográficamente?

Como suele ser imposible (económica y materialmente hablando) recoger información de toda la población objetivo, se recurre a analizar una [muestra](https://economipedia.com/definiciones/muestra-estadistica.html) representativa.

Es decir, en lugar de aplicar [encuestas](https://economipedia.com/definiciones/encuesta.html) o análisis a todas las personas que forman parte del púbico objetivo, se selecciona a un subgrupo (la muestra) sobre el cual se desarrollará el estudio.

La idea es que la muestra refleje lo mejor posible las características de la población. Es decir, deberían compartir datos similares a los que muestra la población en indicadores como edad promedio, nivel de ingresos, porcentaje de hombres y mujeres, entre otros.

# **CAPÍTULO II – MARCO TEORICO**

# **2.1 INTRODUCCIÓN**

# **2.2 SISTEMA DE INFORMACIÓN**

Un sistema de información (SI) es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su uso posterior, generados para cubrir una necesidad u objetivo.

# **2.3 HERRAMIENTAS PARA EL DESARROLLO DE SOFTWARE**

**2.3.1 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.

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.

**2.3.2 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. (developer, 2022)

**2.3.3 JAVASCRIPT**

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 dinámico.

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 será nuestra herramienta principal para diseñar y darle vida a la página, con esto lograremos que sea una página intuitiva y atrayente.

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.

Por último, usaremos JavaScript para agregar las funciones(alertas) de la página web.

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.

# **2.4 HERRAMIENTAS PARA LA IMPLEMENTACION DEL 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.

# **2.4.1 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.

El código PHP suele ser procesado en un [servidor web](https://es.wikipedia.org/wiki/Servidor_web) por un [intérprete](https://es.wikipedia.org/wiki/Int%C3%A9rprete_(inform%C3%A1tica)) PHP implementado como un [módulo](https://es.wikipedia.org/wiki/Complemento_(inform%C3%A1tica)), un [daemon](https://es.wikipedia.org/wiki/Daemon_(inform%C3%A1tica)" \o "Daemon (informática)) o como un ejecutable de [interfaz de entrada común](https://es.wikipedia.org/wiki/Interfaz_de_entrada_com%C3%BAn) (CGI). En un servidor web, el resultado del código PHP interpretado y ejecutado —que puede ser cualquier tipo de datos, como el [HTML](https://es.wikipedia.org/wiki/HTML) generado o datos de imágenes binarias— formaría la totalidad o parte de una respuesta [HTTP](https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto). Existen diversos sistemas de plantillas, [sistemas de gestión de contenidos](https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos) y [frameworks](https://es.wikipedia.org/wiki/Framework_para_aplicaciones_web) que pueden emplearse para organizar o facilitar la generación de esa respuesta. Por otra parte, PHP puede utilizarse para muchas tareas de programación fuera del contexto de la web, como [aplicaciones gráficas](https://es.wikipedia.org/wiki/Interfaz_gr%C3%A1fica_de_usuario) autónomas[7](https://es.wikipedia.org/wiki/PHP#cite_note-8)​ y el control de [drones](https://es.wikipedia.org/wiki/Dron).[8](https://es.wikipedia.org/wiki/PHP#cite_note-9)​ También se puede interpretar y ejecutar un código PHP cualquiera a través de una [interfaz de línea de comandos](https://es.wikipedia.org/wiki/Interfaz_de_l%C3%ADnea_de_comandos) (CLI).

El intérprete estándar de PHP, impulsado por [Motor Zend](https://es.wikipedia.org/wiki/Motor_Zend), es un [software libre](https://es.wikipedia.org/wiki/Software_libre) publicado bajo [Licencia PHP](https://es.wikipedia.org/wiki/Licencia_PHP). PHP ha sido ampliamente portado y puede ser desplegado en la mayoría de los servidores web en casi todos los [sistemas operativos](https://es.wikipedia.org/wiki/Sistema_operativo) y [plataformas](https://es.wikipedia.org/wiki/Plataforma_(inform%C3%A1tica)), de forma gratuita.

# **2.4.2 SGBD**

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.

Una **base de datos no relacional** es un sistema de almacenamiento de datos que cuenta con particularidades que las diferencian del otro gran grupo de bases de datos, esto es, las relacionales. En este artículo vemos en qué consiste una base de datos relacional y cuáles son sus principales características y ventajas.

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.

**2.4.3 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,[1](https://es.wikipedia.org/wiki/MySQL#cite_note-1)​[2](https://es.wikipedia.org/wiki/MySQL#cite_note-2)​ 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) 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), empresa [finlandesa](https://es.wikipedia.org/wiki/Finlandia) desarrolladora del motor [InnoDB](https://es.wikipedia.org/wiki/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)) denominado [MariaDB](https://es.wikipedia.org/wiki/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](https://es.wikipedia.org/wiki/Wikipedia),[5](https://es.wikipedia.org/wiki/MySQL#cite_note-5)​ [Google](https://es.wikipedia.org/wiki/Google)[6](https://es.wikipedia.org/wiki/MySQL#cite_note-mysqlatgoogle-6)​[7](https://es.wikipedia.org/wiki/MySQL#cite_note-7)​ (aunque no para búsquedas), [Facebook](https://es.wikipedia.org/wiki/Facebook),[8](https://es.wikipedia.org/wiki/MySQL#cite_note-mysqlatfacebook-8)​[9](https://es.wikipedia.org/wiki/MySQL#cite_note-9)​[10](https://es.wikipedia.org/wiki/MySQL#cite_note-10)​ [Twitter](https://es.wikipedia.org/wiki/Twitter),[11](https://es.wikipedia.org/wiki/MySQL#cite_note-11)​ [Flickr](https://es.wikipedia.org/wiki/Flickr),[12](https://es.wikipedia.org/wiki/MySQL#cite_note-12)​ y [YouTube](https://es.wikipedia.org/wiki/YouTube).[13](https://es.wikipedia.org/wiki/MySQL#cite_note-13)​

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.

# **2.4.4 OTRAS HERRAMIENTAS ADICIONALES**

Para poder optimizar el trabajo se optará por usar una aplicación stack que contendrá MySQL y apache para poder correr la página web.

**WAMP** es 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.

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.

# **2.4.5 FRAMEWORKS**

# **2.4.6 BOOTSTRAP**

Bootstrap es un “framework de CSS” de código abierto, en otras palabras, es un conjunto de archivos CSS que se incluye de una página o también podemos descargar los archivos y empezar a maquetar el sitio web en minutos sin la necesidad de tocar ninguna línea de css. Bootstrap nos brinda toda la carga de diseño de nuestra página web y a la vez otorgando elegancia gracias a sus clases predefinidas.

# **2.4.7 PATRON DE DISEÑO MVC**

Durante toda la década del setenta, SmallTalk y algunos otros lenguajes como Simula I, fueron construyendo gradualmente el paradigma de programación orientada a objetos y estableciendo conceptos tales como objetos, clases, encapsulación, herencia y polimorfismo.

Si bien dichos lenguajes no son usados actualmente para implementar aplicaciones comerciales, los conceptos que dejaron en el mundo del desarrollo de software están vigentes en la actualidad y son la base de lenguajes modernos como C++, Java o C#.

SmallTalk también fue el primer lenguaje de programación que permitió diseñar interfaces de usuario con múltiples "ventanas" desplegadas en una misma pantalla, concepto que después fue aplicado por GEMS, Macintosh, X11, Windows y otras interfaces gráficas de usuario modernas. El concepto central detrás de las librerías de interfaz de usuario provistas por SmallTalk está basado en el patrón de diseño MVC, creado por el profesor Trygve Reenskaug.

MVC es un patrón de diseño que considera dividir una aplicación en tres módulos claramente identificables y con funcionalidad bien definida: El Modelo, las Vistas y el Controlador.

El modelo es un conjunto de clases que representan la información del mundo real que el sistema debe procesar, así por ejemplo un sistema de administración de datos climatológicos tendrá un modelo que representará la temperatura, la humedad ambiental, el estado del tiempo esperado, etc. sin tomar en cuenta ni la forma en la que esa información va a ser mostrada ni los mecanismos que hacen que esos datos estén dentro del modelo, es decir, sin tener relación con ninguna otra entidad dentro de la aplicación.

El modelo desconoce la existencia de las vistas y del controlador. Ese enfoque suena interesante, pero en la práctica no es aplicable pues deben existir interfaces que permitan a los módulos comunicarse entre sí, por lo que SmallTalk sugiere que el modelo en realidad esté formado por dos submódulos: El modelo del dominio y el modelo de la aplicación. El presente artículo utiliza la propuesta de SmallTalk por ser la base de la implementación de la librería Swing que es vista más adelante.

Se podría decir que el modelo del dominio (o el modelo propiamente dicho) es el conjunto de clases que un ingeniero de software modela al analizar el problema que desea resolver; así, pertenecerían al modelo del dominio: El cliente, la factura, la temperatura, la hora, etc. El modelo del dominio no debería tener relación con nada externo a la información que contiene.

El modelo de la aplicación es un conjunto de clases que se relacionan con el modelo del dominio, que tienen conocimiento de las vistas y que implementan los mecanismos necesarios para notificar a éstas últimas sobre los cambios que se pudieren dar en el modelo del dominio. El modelo de la aplicación es llamado también coordinador de la aplicación.

Las vistas son el conjunto de clases que se encargan de mostrar al usuario la información contenida en el modelo. Una vista está asociada a un modelo, pudiendo existir varias vistas asociadas al mismo modelo; así, por ejemplo, se puede tener una vista mostrando la hora del sistema como un reloj analógico y otra vista mostrando la misma información como un reloj digital.

Una vista obtiene del modelo solamente la información que necesita para desplegar y se actualiza cada vez que el modelo del dominio cambia por medio de notificaciones generadas por el modelo de la aplicación.

El controlador es un objeto que se encarga de dirigir el flujo del control de la aplicación debido a mensajes externos, como datos introducidos por el usuario u opciones del menú seleccionadas por él. A partir de estos mensajes, el controlador se encarga de modificar el modelo o de abrir y cerrar vistas. El controlador tiene acceso al modelo y a las vistas, pero las vistas y el modelo no conocen de la existencia del controlador.

# **2.4.8 VISUAL STUDIO CODE**

Visual Studio Code es un [editor de código fuente](https://es.wikipedia.org/wiki/Editor_de_c%C3%B3digo_fuente) desarrollado por [Microsoft](https://es.wikipedia.org/wiki/Microsoft) para [Windows](https://es.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://es.wikipedia.org/wiki/Linux), [macOS](https://es.wikipedia.org/wiki/MacOS) y [Web](https://es.wikipedia.org/wiki/World_Wide_Web). Incluye soporte para la [depuración](https://es.wikipedia.org/wiki/Depuraci%C3%B3n_de_programas), control integrado de [Git](https://es.wikipedia.org/wiki/Git), resaltado de sintaxis, finalización inteligente de código, [fragmentos](https://es.wikipedia.org/wiki/Snippet) y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de [código abierto](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto), aunque la descarga oficial está bajo [software privativo](https://es.wikipedia.org/wiki/Software_privativo) e incluye características personalizadas por [Microsoft](https://es.wikipedia.org/wiki/Microsoft).

Visual Studio Code se basa en [Electron](https://es.wikipedia.org/wiki/Electron_(software)), un framework que se utiliza para implementar [Chromium](https://es.wikipedia.org/wiki/Chromium_(navegador)) y [Node.js](https://es.wikipedia.org/wiki/Node.js) como aplicaciones para escritorio, que se ejecuta en el motor de diseño Blink. Aunque utiliza el framework Electron, el software no usa [Atom](https://es.wikipedia.org/wiki/Atom_(editor_de_textos)) y en su lugar emplea el mismo componente editor (Monaco) utilizado en Visual Studio Team Services (anteriormente llamado Visual Studio Online).

# **2.4.9 PHP MAILER**

La librería PHPMailer nos permite, entre otras funcionalidades, establecer una conexión SMTP con un servidor de correo. Este servidor SMTP será quién envíe realmente nuestro correo electrónico.

# **2.4.10 HOSTING Y DOMINIO**

ISPConfig es un software que permite fácilmente administrar servidores, espacio-web, revendedores y simples hosts virtuales.

**ISPConfig consiste de 3 niveles:**

ISPConfig para administradores de servidores, permite la administración de servidores, espacio-web, revendedores y clientes.

ISPConfig para revendedores, permite la administración de espacio-web, clientes y virtuales hosts (sitios).

ISPConfig para webmasters, permite la administración de sitios, sub-dominios y buzones de correo.

# **2.5 METODOLOGIAS ÁGILES**

# **2.5.1 DEFINICION DE METODOLOGIAS AGILES**

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

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.

# **2.5.2 PRINCIPIOS DEL MANIFIESTO AGIL**

En marzo de 2001 diecisiete críticos de los modelos de mejora del desarrollo de software basados en procesos, convocados por Kent Benck, quien había publicado un par de años Extreme Programming Explained, libro en el que exponía una nueva metodología denominada Extrem Pramming, se reunieron en Salt Lake City para tratar sobre técnicas y procesos para desarrollar software.

En la reunión se acuño el término “Métodos Agiles” para definir a los métodos que estaban surgiendo como alternativa a las metodologías formales como CMMI y SPICE a las que se consideraba excesivamente pesadas y rígidas por su carácter normativo y fuerte dependencia de planificaciones detalladas previas al desarrollo.Los integrantes de la reunión resumieron los principios sobre los que se basan los métodos alternativos en cuatro postulados lo que ha quedado denominado como Manifiesto Ágil.

**El manifiesto Ágil está fundamentado en los siguientes valores:**

* Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. Las personas son el principal factor de éxito de un proyecto software. Es más importante construir un buen equipo de trabajo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que este configure su propio entorno de desarrollo en base a sus necesidades.
* Desarrollar software que funciona más que conseguir una buena documentación. La regla a seguir es no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante. Estos documentos deben ser cortos y centrarse en lo fundamental.
* La colaboración con el cliente más que la negociación de un contrato. Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.

**LOS PRINCIPIOS DEL MANIFIESTO AGIL**

* La prioridad es satisfacer al cliente mediante tempranas y continuas entregas que le aporte un valor
* Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva
* Entregar frecuentemente software que función desde un par de semanas a un par de meses con el menor intervalo de tiempo posible entre entregas
* La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto
* Construir el proyecto entorno a los individuos motivados. Darles el entorno y apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
* El dialogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo
* El software que funciona es la medida principal de progreso
* Los procesos agiles promueven un desarrollo sostenible
* La atención continua a la calidad técnica y al buen diseño mejora la agilidad
* La simplicidad es lo esencial
* Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos
* En intervalos regulares el equipo reflexiona respecto a cómo llegar a ser más eficiente y según esto se ajusta el comportamiento

# **2.5.3 CARACTERISTICAS DE LAS METODOLOGIAS AGILES**

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.

# **2.5.4 METODOLOGIAS AGILES VS TRADICIONALES**

Las diferencias que podemos encontrar entre estas 2 metodologías son las siguientes:

**FLEXIBILIDAD**

La gestión de proyectos tradicional prácticamente no deja márgenes para realizar cambios en el producto. Es un proceso rígido, en el cual una vez que el plan ha finalizado, los gerentes se encargan de comunicárselo a sus equipos y se aseguran de que cada uno de los integrantes lo lleve a cabo de la mejor forma posible. Hay mucha resistencia frente a cualquier cambio que se quiera realizar, debido a que puede interrumpir la programación del proyecto.

A diferencia de la gestión de proyectos tradicional, la metodología Agile ofrece mucha más flexibilidad en lo que se refiere a realizar cambios. En este caso, los miembros del equipo son libres de dar a conocer e intercambiar ideas que crean que puede ayudar a mejorar aún más el producto, es por eso que se les permite experimentar y descubrir/dar con las mejores alternativas posibles. El método Agile se enfoca más en obtener el producto correcto que en seguir estructuras rígidas.

**APROPIACION Y TRANSPARENCIA**

Otra de las diferencias a tener en cuenta entre estas metodologías es que, en la gestión de proyectos tradicional, la propiedad le pertenece al director de Proyectos, por lo tanto, es su responsabilidad planear y documentar todo el recorrido del producto.

Aparte de los Gerentes, sólo los clientes están involucrados/forman parte de la fase de planificación, pero una vez que la implementación comienza, su participación es nula.

Dado que los gerentes llevan las riendas del proyecto, los miembros del equipo generalmente no tienen voz en el resultado de sus esfuerzos o en el progreso del proyecto.

Mientras que en la metodología Agile, los proyectos son propiedad de los integrantes del equipo de trabajo. Todos ellos colaboran en conjunto para idear un plan diseñado para finalizar el trabajo dentro del tiempo y costo estimado. Además, se les permite conocer el progreso del producto, desde el comienzo hasta el final del mismo. Toda esta transparencia, desempeña un papel importante en el mantenimiento de un entorno laboral productivo y altamente comprometido.

**SOLUCION DE PROBLEMAS**

En caso de aparecer obstáculos inesperados, los miembros del equipo de trabajo deben comunicar el asunto a sus gerentes. De todas formas, transmitir los problemas cada vez que surgen no es una opción viable/factible, ya que esto puede causar retrasos que implican exceder el tiempo límite, además de aumentar el costo total.

Dentro de las [metodologías ágiles,](https://robertotouza.com/agile/#Metodologias_Agiles) los equipos tienen la autoridad para tomar las decisiones por sí mismos. De esta forma, intenta resolver los problemas internamente y evitan pérdidas de tiempo innecesarias. Al estar comprometidos y cerca del proceso, sus conocimientos ayudan a abordar la mayoría de los problemas que dificultan su progreso. Por lo tanto, a menos que haya necesidad de tomar decisiones extremas, los miembros del equipo difícilmente eleven asuntos triviales/sin importancia a sus gerentes.

**ALCANCE DE RETROALIMENTACION**

En la gestión de proyectos tradicional, cada proceso es claramente definido y planeado desde el comienzo. Éste debe ser finalizado y completo dentro del tiempo y costo estimado, lo que significa que cualquier cambio brusco/grande/de importancia o feedback que pueda retrasar la fecha límite no se considera/ se deja de lado. Mientras que dentro de los métodos Agile se permite el feedback constante, el cual es muy útil para proporcionar un mejor rendimiento.

Debido a la alta aceptación de la retroalimentación en la metodología ágil, ésta se ha convertido en la primera opción de muchos gerentes de proyectos y desarrolladores de software, ya que con el uso de estas metodologías pueden responder a los requerimientos/pedidos de los clientes a medida que éstos a su vez validan cada iteración de modo tal que les permita entregar/enviar productos o servicios de alta calidad dentro del plazo de entrega/dentro del periodo de tiempo asignado.

# **2.6 METODOLOGIA SCRUM**

# **2.6.1 HISTORIA**

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.

# **2.6.2 FORMA DE TRABAJO**

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.

# **2.6.3 CICLO DE VIDA**

* **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.

**2.6.3.1 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.

# **2.6.4 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).
* **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

# **2.6.5 CARACTERISTICAS**

* **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.

# **2.6.6 VENTAJAS**

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”.

# **2.6.7 DESVENTAJAS**

* **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.

# **2.7 FUNDAMENTOS TEORICOS DEL SISTEMA**

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

# **3.1 INTRODUCCION**

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

# **3.2 REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES**

# **3.2.1 REQUERIMIENTOS FUNCIONALES**

* **RF0.** Herramientas, el sistema debe permitir que los clientes busquen un producto por nombre o descripción mediante un filtro de búsqueda.
* **RF1.** Flujo de trabajo, al realizar un pedido, la solicitud pasará al siguiente paso del flujo de trabajo.
* **RF2.** Seguridad, el sistema permitirá a los usuarios autorizados ingresar productos y modificarlos para posteriormente realizar la exposición en la página web.
* **RF3.** Compatibilidad, la aplicación debe poder utilizarse sin necesidad de instalar ningún software adicional y simplemente contar con un navegador web (Firefox, Chrome, Edge, Opera).
* **RF4.** Gestión de control, los permisos de acceso al sistema podrán ser cambiados por el administrador de la página.

# **3.2.2 REQUERIMIENTOS NO FUNCIONALES**

* **RNF0.** Rendimiento, el tiempo para iniciar o reiniciar el sistema no podrá ser mayor a 5 minutos.
* **RNF1.** Interfaz, la interfaz de usuario será implementada para navegadores web únicamente con (HTML, Php, Css, JavaScript y Bootstrap).
* **RNF2**. Interfaz, la interfaz del sistema tanto como para el cliente y administrador debe ser intuitiva y llamativa.
* **RNF3.** Compatibilidad, la aplicación debe poder utilizarse en los navegadores web más usados actualmente como: Chrome, Firefox y Edge.
* **RNF4.** Aprendizaje, el tiempo de aprendizaje del sistema por un usuario deberá ser menor a 4 horas.
* **RNF5.** Disponibilidad, el sistema deberá estar disponible las 24horas/7días siempre que los servidores donde se albergan estén funcionando correctamente o no tengan ningún riesgo inconveniente.
* **RNF6.** Mantenibilidad, la mantenibilidad del sitio queda en manos del administrador del sistema, para lo cual se capacitará a esta persona. Si el mantenimiento va más allá de las manos del administrador entonces se contactará al desarrollador del sistema.

# **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:** No urgente (pero importante) | |
| **Programador Responsable:** Pinto Mora Victor Angel | |
| **Descripción:** Como cliente solicito que la interfaz gráfica sea intuitiva para poder ver y realizar los pedidos sin ningún problema y de la manera más rápida. | |
| **Observaciones:** Los colores son los solicitados de la empresa | |

|  |  |
| --- | --- |
| **HISTORIAS DE USUARIO** | |
| **Numero:** 2 | **Usuario:** Vendedor |
| **Nombre de la Historia:** Administración de Productos | |
| **Prioridad:** Urgente (Importante) | |
| **Programador Responsable:** Pinto Mora Victor Angel | |
| **Descripción:** Como vendedor quiero poder administrar y modificar los productos para poder tenerlos organizados de la mejor manera y mantener actualizada la página. | |
| **Observaciones:** Ninguna | |

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

|  |  |
| --- | --- |
| **HISTORIAS DE USUARIO** | |
| **Numero:** 4 | **Usuario:** Cliente |
| **Nombre de la Historia:** Descripciones Técnicas y Físicas de un Producto | |
| **Prioridad:** Urgente (No importante) | |
| **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:** 5 | **Usuario:** Vendedor |
| **Nombre de la Historia:** Administración de Pedidos de los Clientes | |
| **Prioridad: Urgente (Importante)** | |
| **Programador Responsable:** Pinto Mora Victor Angel | |
| **Descripción:** Como Vendedor 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:** 6 | **Usuario:** Cliente |
| **Nombre de la Historia:** AgregarCarrito de compras y pedidos | |
| **Prioridad:** Urgente (Importante) | |
| **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 junto al total de la suma de todos los productos agregados. | |
| **Observaciones:** Ninguna | |

|  |  |
| --- | --- |
| **HISTORIAS DE USUARIO** | |
| **Numero:** 7 | **Usuario:** Vendedor |
| **Nombre de la Historia:** Estados de un Pedido | |
| **Prioridad:** Urgente (Importante) | |
| **Programador Responsable:** Pinto Mora Victor Angel | |
| **Descripción:** Como Vendedor solicito que cada pedido contenga un campo donde podamos asignar un estado (Completo, Aprobado, Cancelado, Reembolsado, Pendiente, Entregado) con el objetivo de poder administrarlos de mejor manera y así el cliente tenga conocimiento sobre el estado del producto que solicito. | |
| **Observaciones:** Ninguna | |

|  |  |
| --- | --- |
| **HISTORIAS DE USUARIO** | |
| **Numero:** 8 | **Usuario:** Vendedor |
| **Nombre de la Historia:** Modificación de Clientes | |
| **Prioridad:** Urgente (Importante) | |
| **Programador Responsable:** Pinto Mora Victor Angel | |
| **Descripción:** Como Vendedor solicito poder realizar modificaciones sobre los clientes registrador y también poder crear nuevos clientes para poder registrarlos en el sistema web. | |
| **Observaciones:** Ninguna | |

|  |  |
| --- | --- |
| **HISTORIAS DE USUARIO** | |
| **Numero:** 9 | **Usuario:** Supervisor |
| **Nombre de la Historia:** Administración de Categorías | |
| **Prioridad:** Urgente (Importante) | |
| **Programador Responsable:** Pinto Mora Victor Angel | |
| **Descripción:** Como supervisor de la página web solicito poder crear y modificar categorías, con el fin de ordenar la página y productos. | |
| **Observaciones:** Ninguna | |

|  |  |
| --- | --- |
| **HISTORIAS DE USUARIO** | |
| **Numero:** 10 | **Usuario:** Cliente |
| **Nombre de la Historia:** Pedidos | |
| **Prioridad:** Urgente (Importante) | |
| **Programador Responsable:** Pinto Mora Victor Angel | |
| **Descripción:** Como cliente de la página web solicito poder ver la lista de los todos los pedidos que realice con detalles exactos sobre el pedido, poder ver el total, producto solicitado, fecha y tipo de pago que realice. | |
| **Observaciones:** Ninguna | |

|  |  |
| --- | --- |
| **HISTORIAS DE USUARIO** | |
| **Numero:** 11 | **Usuario:** Administrador |
| **Nombre de la Historia:** Reportes | |
| **Prioridad:** Urgente (Importante) | |
| **Programador Responsable:** Pinto Mora Victor Angel | |
| **Descripción:** Como administrador solicito que la pagina realice reportes (Excel, Pdf, Csv) con el fin de contener la información que será de vital importancia para la empresa. | |
| **Observaciones:** Las acciones de reportes estará habilitada tanto para todos los roles. | |

# **3.3.2 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 | Al realizar el diseño de la página se debe tomar en cuenta los colores de la empresa | Primer Sprint |
| 2 | Administración de Productos | 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 | Descripciones Técnicas y Físicas de un Producto | 04/08/2022 | SI | Campo de texto para ingresar características antes de la publicación del producto. | Segundo Sprint |
| 5 | Administración de Pedidos de los Clientes | 04/08/2022 | SI |  | Segundo Sprint |
| 6 | AgregarCarrito de compras y pedidos | 25/08/2022 | SI | El carrito debe poner los productos junto a su imagen | Tercer Sprint |
| 7 | Estados de un Pedido | 25/08/2022 | SI |  | Tercer Sprint |
| 8 | Modificación de Clientes | 25/08/2022 | SI |  | Tercer Sprint |
| 9 | Administración de Categorías | 25/08/2022 | SI |  | Tercer Sprint |
| 10 | Pedidos | 25/08/2022 | SI |  | Tercer Sprint |
| 11 | Reportes | 25/08/2022 | SI | Las acciones de reportes estarán habilitadas para todos los roles. | Tercer Sprint |

# **3.3.4 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 | * El personal debe poder agregar las categorías o eliminarlas * La modificación solo debe ser para un rol. |
| Tercer Sprint | 25/08/2022 | 8/09/2022 | 08/09/2022 | * Agregar opción de compra en el carrito |

# **3.3.5 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.

# **3.3.6 PLANIFICACION 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 Productos | 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 | Descripciones Técnicas y Físicas de un Producto | 04/08/2022 | 25/08/2022 | SI |  |
| 5 | Segundo Sprint | Administración de Pedidos de los Clientes | 4/08/2022 | 25/08/2022 | SI |  |
| 6 | Tercer Sprint | AgregarCarrito de compras y pedidos | 25/08/2022 | 08/09/2022 | SI |  |
| 7 | Tercer Sprint | Estados de un Pedido | 25/08/2022 | 08/09/2022 | SI |  |
| 8 | Tercer Sprint | Modificación de clientes | 25/08/2022 | 8/09/2022 | SI |  |
| 9 | Tercer Sprint | Administración de Categorías | 25/08/2022 | 08/09/2022 | SI |  |
| 10 | Tercer Sprint | Pedidos | 25/08/2022 | 08/09/2022 | SI |  |
| 11 | Tercer Sprint | Reportes | 25/08/2022 | 08/09/2022 | SI |  |

# **3.4 DISEÑO DE LA BASE DE DATOS**

# **3.4.1 DESCRIPCION 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 página web los atributos de los clientes pueden ser nombres, direcciones y numero de contado y demás.

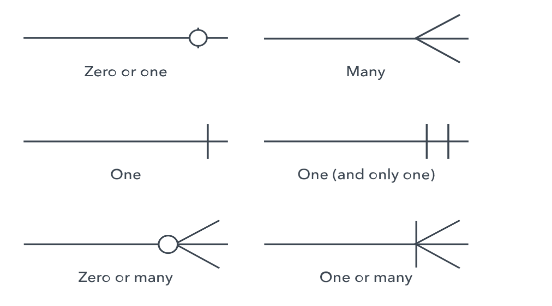
Las relaciones nos describen como van a interactuar las entidades entre sí. En las relaciones tenemos varios tipos de relación (Uno a muchos, Uno a uno, Muchos a Muchos).

**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.

# **3.4.2 SIMBOLOGIA Y ELEMENTOS**

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.

# **3.4.3 DIAGRAMA ENTIDAD RELACION DEL PROYECTO**

**Ilustración 2 Diagrama Entidad Relación del Sistema Web**

# **3.4.4 DESCRIPCION DE CADA UNA DE LAS TABLAS DE LA B.D**

**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.

# **3.4.5 DICCIONARIO DE DATOS**

|  |  |  |
| --- | --- | --- |
| **TABLA: PERSONA** | | |
| **Atributo** | **Tipo Dato** | **Descripción** |
| Idpersona | Int | Almacenaremos el id de las personas registradas |
| Identificacion | varchar | Almacenaremos la identificación de cada cliente para mayor seguridad en los pedidos |
| Nombres | Varchar | Almacenaremos los nombres de los clientes registrados |
| Apellidos | Varchar | Almacenar datos de igual forma que los nombres del cliente |
| Telfono | Int | Campo que nos facilitara contactar al cliente |
| Email\_user | Varchar | Correo electrónico para poder registrar y contactar al cliente |
| Password | Varchar | Campo para el código de acceso para el usuario |
| RolId | Int | Campo para asignar las funciones del usuario |
| DateCreated | Datetime | Dato para saber la fecha de creación de cada usuario |
| Status | Int | Campo para saber el estado de la persona registrada |

|  |  |  |
| --- | --- | --- |
| **TABLA: PEDIDO** | | |
| **Atributo** | **Tipo Dato** | **Descripción** |
| Idpedido | Int | Almacenaremos el id de los pedidos existentes |
| Personaid | Int | Ingresaremos el id de la persona que realizo el pedido |
| Fecha | Datetime | Campo para conocer la fecha y hora que se realizó el pedido |
| Costo\_envio | Decimal | Campo para agregar el costo de envió de cada pedido que se realice |
| Monto | Int | Con esto sabremos el total del pedido que realizo el cliente |
| TipoPagoid | Int | Identificaremos el tipo de pago que selecciono el cliente (Efectivo, Transferencia) |
| direccionEnvio | Text | Campo para recibir la información sobre la dirección del cliente |
| Status | Varchar | Podremos ver el estado del pedido |

|  |  |  |
| --- | --- | --- |
| **TABLA: PRODUCTO** | | |
| **Atributo** | **Tipo Dato** | **Descripción** |
| Idproducto | Int | Almacenaremos el id de los productos registrados en la pagina |
| Categoriaid | Int | Campo para saber a qué categoría pertenecerá cada producto |
| Código | Varchar | Código del producto para diferenciarlos |
| Nombre | Varchar | Campo para los nombres de los productos registrados en la pagina |
| Descripción | Varchar | Podremos ingresar un campo con una descripción técnica para cada producto |
| Stock | Int | Sabremos la cantidad existente en nuestra base de datos de cada producto |
| Imagen | Img | Campo para agregar una foto del producto |
| Datecreated | Datetime | Con este campo tendremos la información sobre la fecha en la que se registro o creo el producto |
| Status | Int | Identificaremos el estado de cada producto (activo, inactivo) |

|  |  |  |
| --- | --- | --- |
| **TABLA: DETALLE\_PEDIDO** | | |
| **Atributo** | **Tipo Dato** | **Descripción** |
| Id | Int | Identificador para los detalles |
| Pedidoid | Int | Identificador de los pedidos |
| Productoid | Int | Recibiremos el id del producto asociado al pedido |
| Precio | Decimal | Campo para saber el precio de los productos asociados al pedido |
| Cantidad | Int | Este campo nos mostrara la cantidad total |

|  |  |  |
| --- | --- | --- |
| **TABLA: CATEGORIA** | | |
| **Atributo** | **Tipo Dato** | **Descripción** |
| Idcategoria | Int | Almacenaremos el identificador de cada categoría |
| Nombre | Varchar | Campo para ingresar el nombre de las categorías creadas |
| Descripción | Varchar | Agregaremos una breve descripción sobre la información de la categoría |
| Portada | Varchar | Ingresaremos una portada para la categoría creada |
| Datecreated | Datetime | Almacenara la fecha en la que se creó la categoría |
| Status | Int | Al igual que otras tablas este campo será fundamental para poder mostrar o no la categoría mediante estados (activo y no activo) |

|  |  |  |
| --- | --- | --- |
| **TABLA: ROL** | | |
| **Atributo** | **Tipo Dato** | **Descripción** |
| Idrol | Int | Este campo asignara un rol a los usuarios con el fin de dividir funciones |
| Nombrerol | Varchar | Se agregará el nombre del rol |
| Descripción | Varchar | Breve descripción de las funciones que tendrá el rol |
| Portada | Varchar | Ingresaremos una portada para la categoría creada |
| Status | Varchar | Podremos ver el estado de cada rol |

|  |  |  |
| --- | --- | --- |
| **TABLA: TIPOPAGO** | | |
| **Atributo** | **Tipo Dato** | **Descripción** |
| Idtipopago | Int | Campo para asignar un id al tipo de pago que se tenga en la página |
| Tipopago | Varchar | El nombre del tipo de pago |
| Status | Int | Veremos el estado del tipo de pago |

# **3.5 FASE DE CONSTRUCCION**

# **3.5.1 HERRAMIENTA DE IMPLEMENTACION**

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

# **3.5.2 HERRAMIENTAS DE DISEÑO**

# **3.5.2.1 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.

# **3.5.2.2 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.

# **3.5.2.3 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.

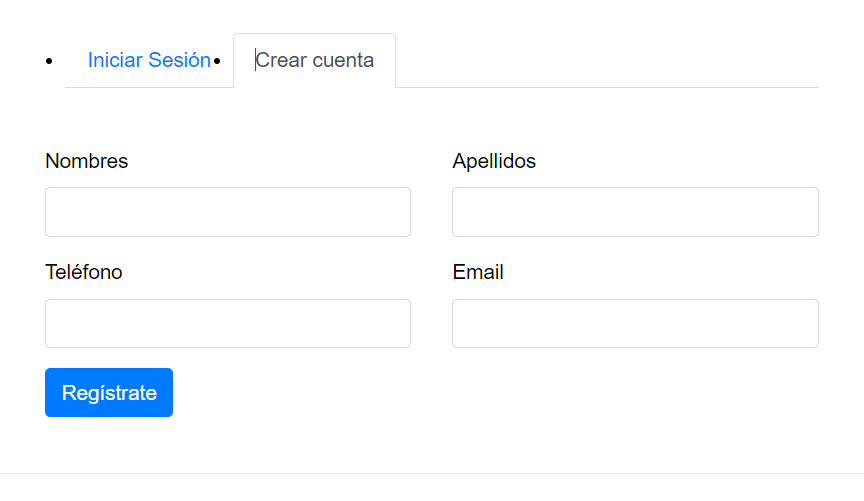
# **3.5.3 ENTORNO DE DESARROLLO**

# **3.5.3.1 IMPLEMENTACION DEL ACCESO AL SISTEMA**

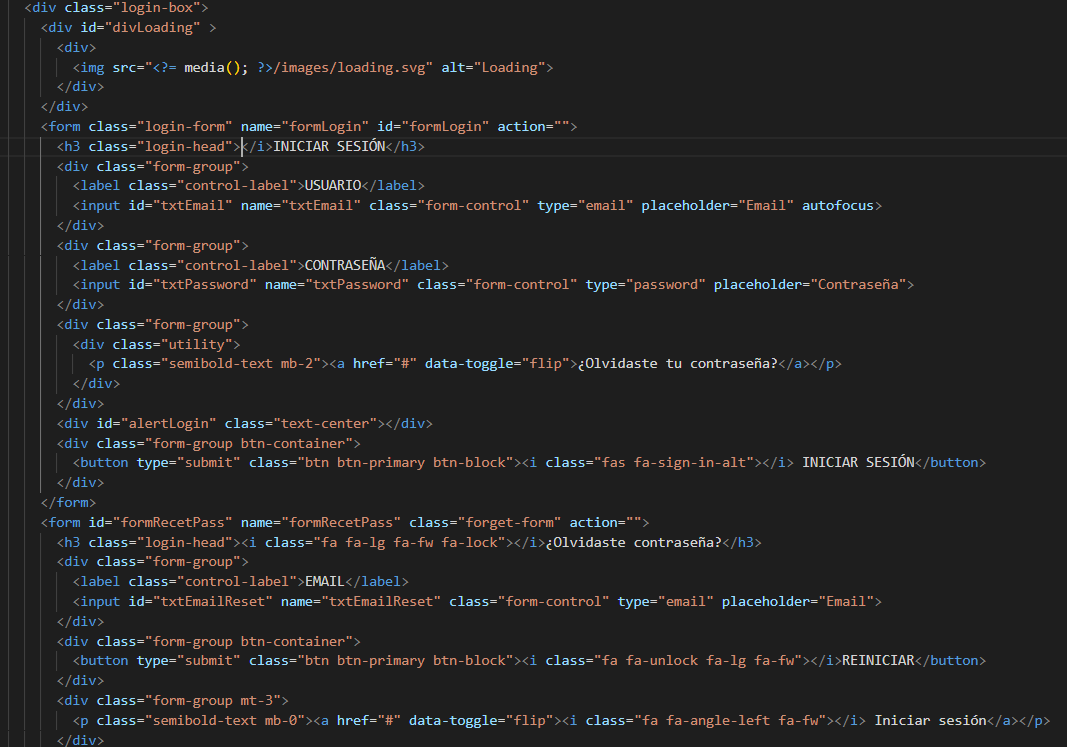
En el primer sprint se llevó a cabo la realización de la interfaz gráfica y login de la página con el objetivo de que sea intuitiva para el cliente y que pueda acceder con un rol a la página tanto como cliente o como administrador.



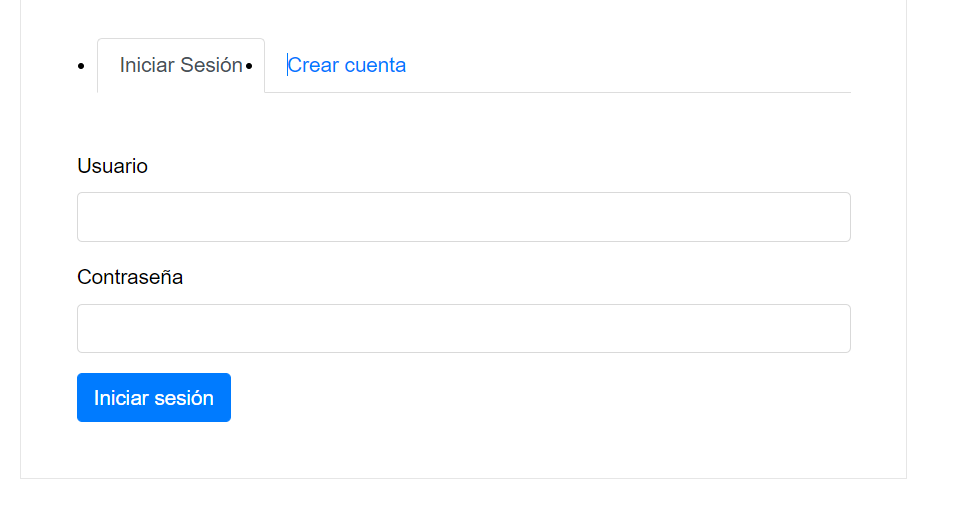
**Ilustración 3 Registro Código**

****

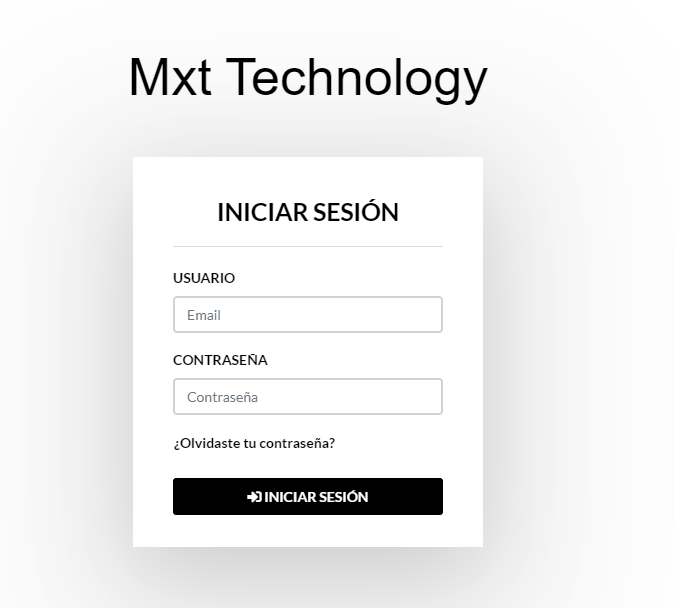
**Ilustración 4 Registro Interfaz Grafica**

****

**Ilustración 5 Login Código**

****

**Ilustración 5 Login para compra. Interfaz Grafica**

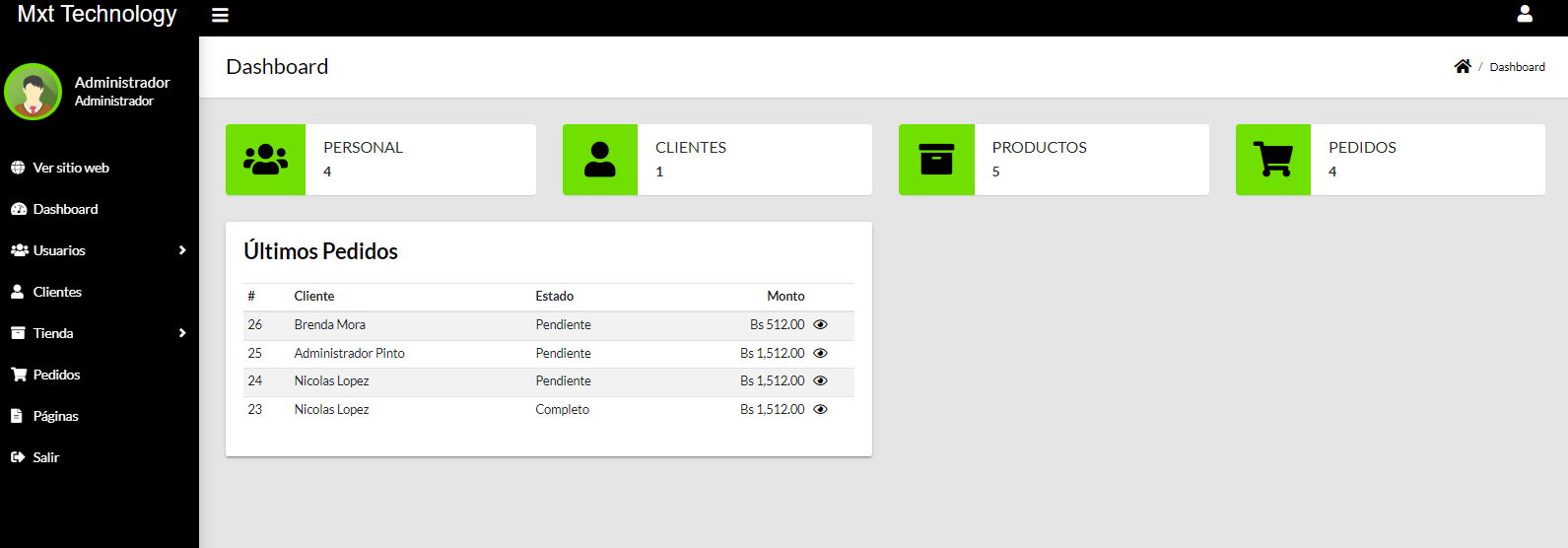


**Ilustración 6 Login principal. Interfaz Grafica**

# **3.5.3.2 IMPLEMENTACION DE LA INTERFAZ GRAFICA**

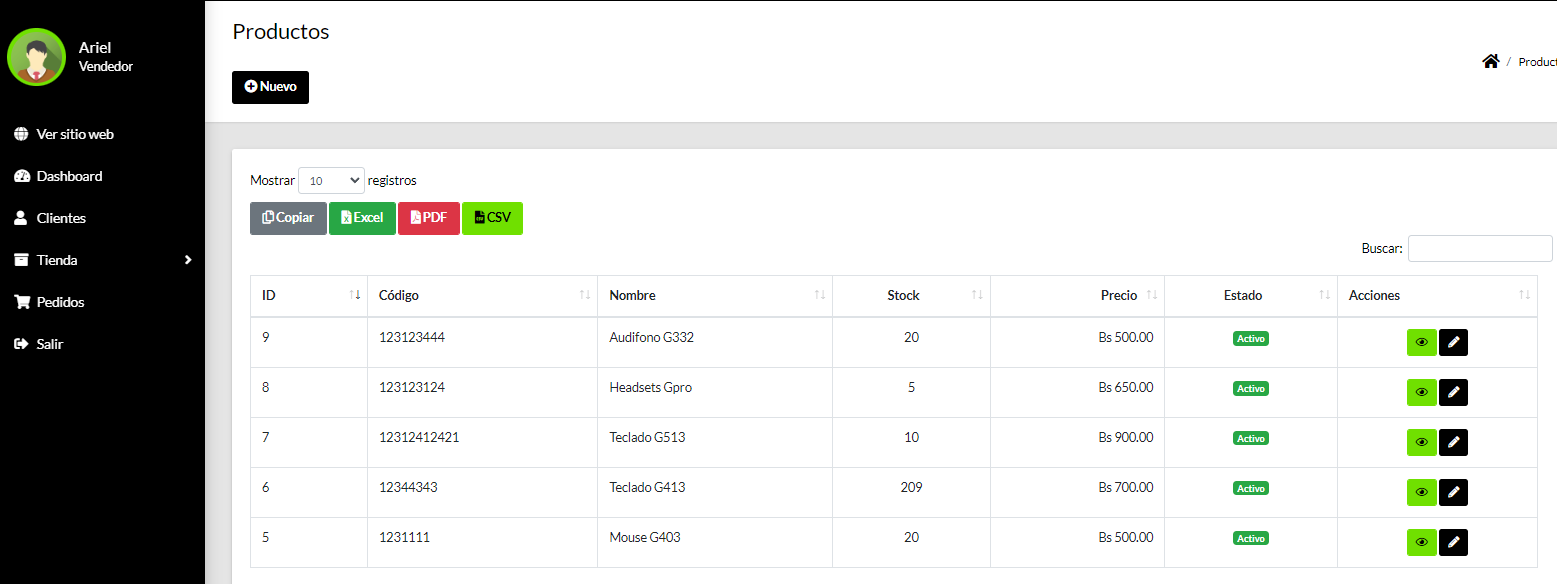


**Ilustración 7 Página principal Interfaz Grafica**

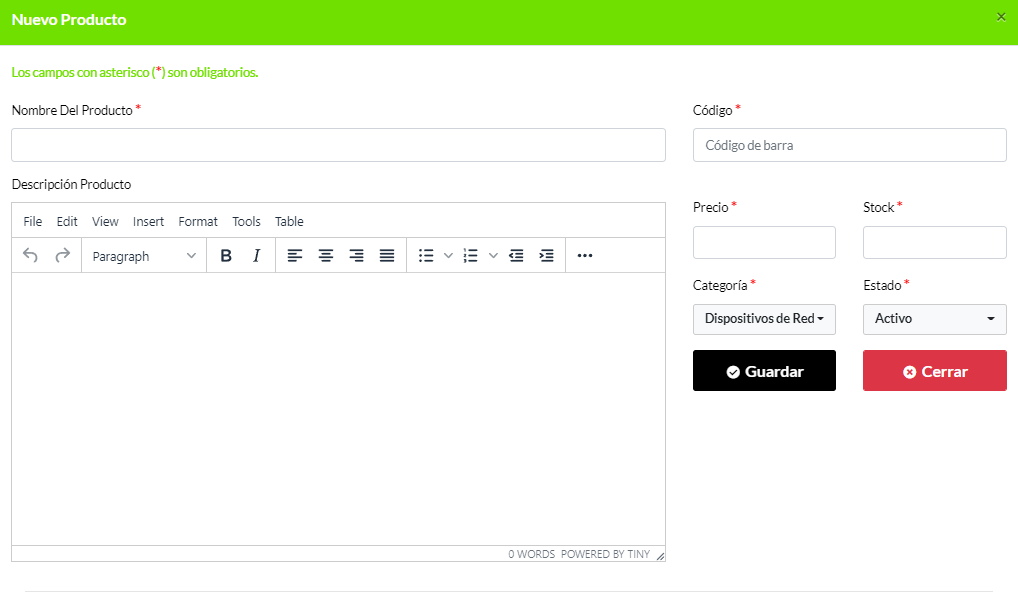


**Ilustración 8 Página administrativa Interfaz Grafica**

# **3.5.3.3 IMPLEMENTACION DE ADMINSITRACION DE PRODUCTOS**



**Ilustración 9 Administración de Productos Interfaz Grafica**

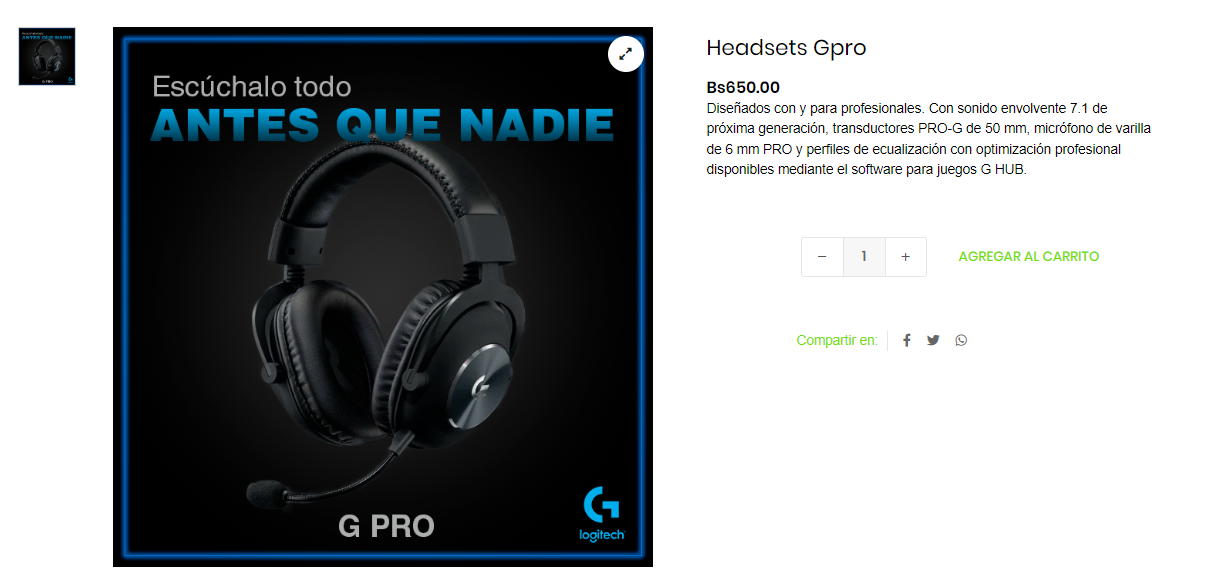


**Ilustración 10 Crear Productos Interfaz Grafica**



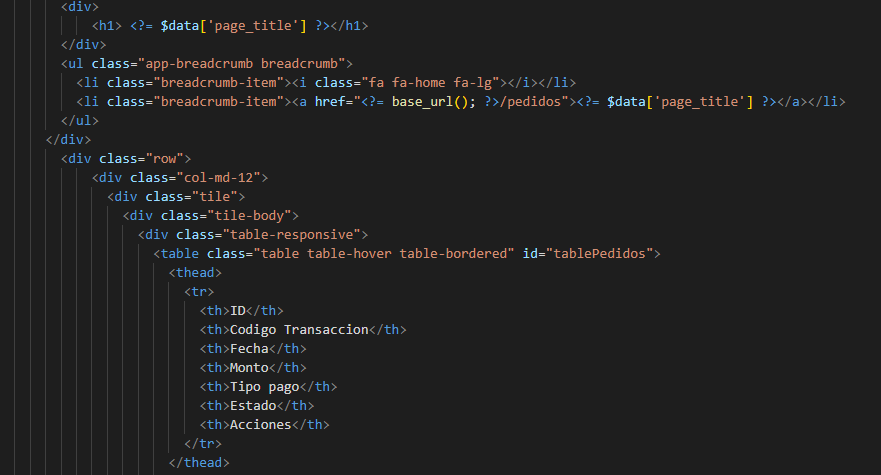
**Ilustración 11 Crear Productos Código**

# **3.5.3.4 IMPLEMENTACION DE DESCRIPCIONES TECNICAS PARA LOS PRODUCTOS**



**Ilustración 12 Descripciones Técnicas de los productos Interfaz Grafica**

# **3.5.3.5 IMPLEMENTACION PARA LA ADMINISTRACION DE PEDIDOS DE LOS CLIENTES**

**Ilustración 13 Administración de pedidos Interfaz Grafica**

**Ilustración 14 Administración de pedidos Código**

# **3.5.3.6 IMPLEMENTACION DE CARRITO DE COMPRAS**

# 

**Ilustración 15 Carrito de Compras Interfaz Grafica**

# **Ilustración 16 Carrito de Compras Código**

# **3.5.3.7 IMPLEMENTACION DE LOS ESTADOS DE LOS PEDIDOS**

**Ilustración 17 Estados de los Pedidos Interfaz Grafica**

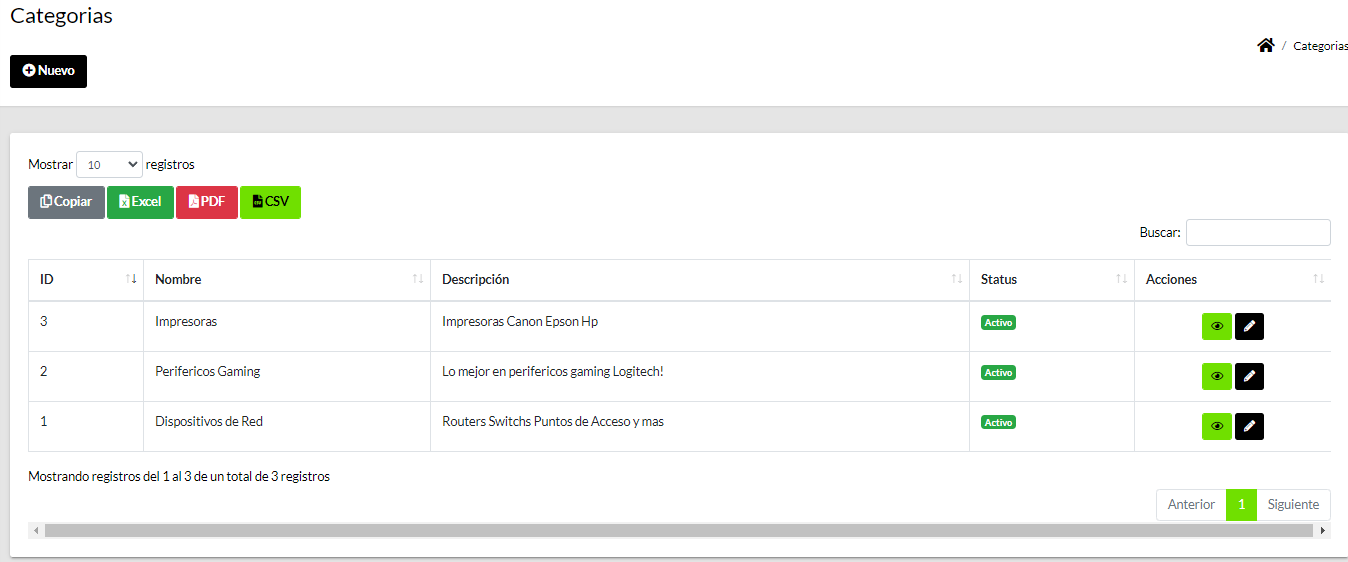
**Ilustración 18 Estados de los Pedidos Código**

# **3.5.3.8 IMPLEMENTACION DE MODIFICACION DE CLIENTES**

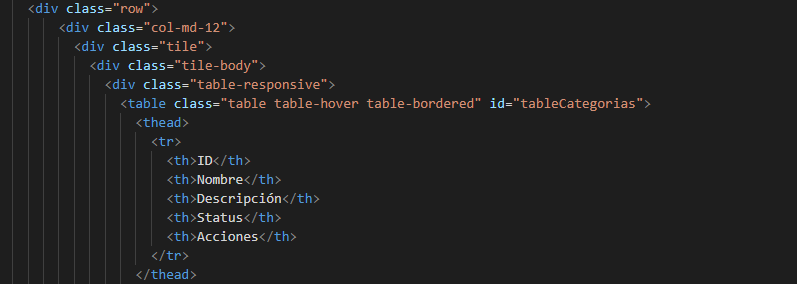
**Ilustración 19 Modificación de Clientes Interfaz Grafica**

**Ilustración 20 Modificación de Clientes Código**

# **3.5.3.9 IMPLEMENTACION DE CATEGORIAS**

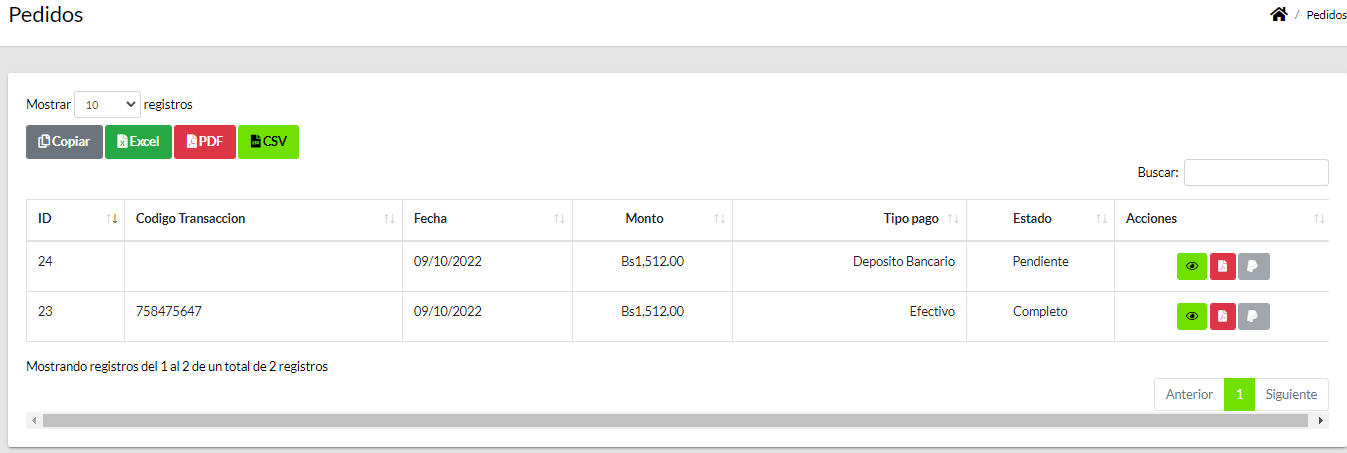


**Ilustración 21 Crear y Modificar Categorías Interfaz Grafica**

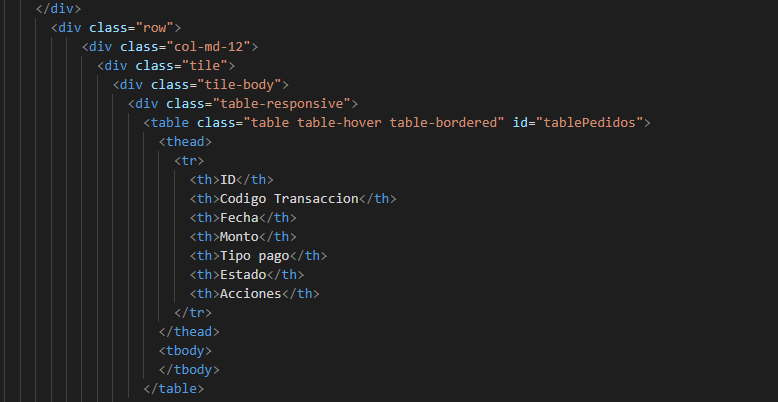


**Ilustración 22 Crear y Modificar Categorías Código**

# **3.5.4. IMPLEMENTACION DE LISTA DE PEDIDOS**



**Ilustración 23 Lista de pedidos para el Cliente Interfaz Grafica**

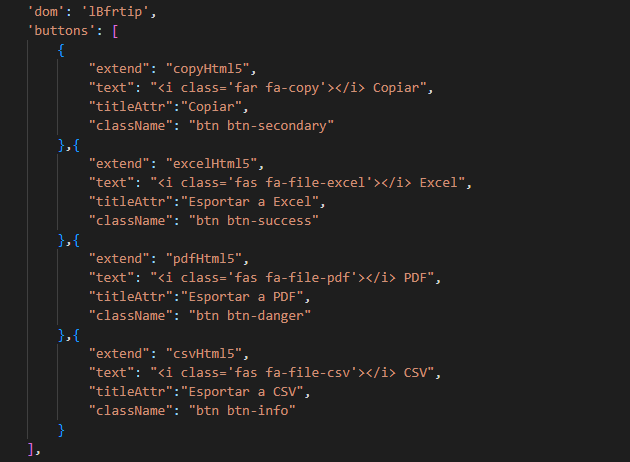


**Ilustración 24 Lista de Pedidos Código**

# **3.5.4.1 IMPLEMENTACION DE REPORTES**



**Ilustración 25 Generador de reportes Interfaz Grafica**



**Ilustración 26 Generador de Reportes JavaScript Código**

# **3.5.4 DIAGRAMA DE ACTIVIDADES 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

# **3.5.5 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 página web y por último la generación de reportes.

# **3.5.6 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 página web.

# **3.5.6.1 REQUERIMIENTOS DE 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 página 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.

# **3.5.6.2 REQUERIMIENTOS 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.

# **3.5.7 INSTALACION Y CONFIGURACION DEL SISTEMA**

# **3.5.7.1 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.

# **3.5.7.2 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.

# **3.5.7.3 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).

# **3.6 CONCLUSIONES Y RECOMENDACIONES**

# **3.6.1 CONCLUSIONES**

# **3.6.2 RECOMENDACIONES**

# **BIBLIOGRAFIA**

# **ANEXOS**

# **MANUAL DE USUARIO**

# **MANUAL TECNICO**

# 