

Universidad de San Carlos de Guatemala.  
Centro Universitario de Occidente.  
División de Ciencias de la Ingeniería.  
Teoría de Sistemas I.  
Ing. Pedro Domingo.



## “DOCUMENTACIÓN TÉCNICA - MARKET CHAPIN”

Diego José Maldonado Monterroso.  
Carné: 201931811.  
Link Repo: <https://github.com/DiegoMaldonado19/MarketChapin>

Quetzaltenango, Guatemala.

05 de septiembre de 2024.

## INTRODUCCIÓN

Este manual técnico está diseñado para guiar en el desarrollo de un sistema web utilizando el modelo de arquitectura MVC (Modelo-Vista-Controlador). Se emplearán tecnologías accesibles y ampliamente utilizadas como PHP, JavaScript, HTML y CSS.

El objetivo de este manual es proporcionar una comprensión clara y detallada de cómo implementar cada componente del modelo MVC, asegurando una separación adecuada de las responsabilidades y facilitando el mantenimiento y escalabilidad del sistema.

A lo largo de este documento, se explorarán las mejores prácticas y patrones de diseño que permitirán construir aplicaciones web robustas y eficientes. Ya sea que el lector esté comenzando en el desarrollo web o buscando mejorar sus habilidades, este manual servirá como una referencia valiosa en el camino hacia la creación de aplicaciones web modernas y funcionales.

## PENSAMIENTO SISTÉMICO

En un mundo donde las tecnologías crecen a pasos agigantados, es nuestro deber como Ingenieros en Sistemas generar soluciones que se adapten a las necesidades de nuestros clientes, así como a su contexto. Si viajamos al pasado, ha existido un modelo que ha sobrevivido y se ha adaptado a estos nuevos paradigmas este es el sistema de compra-venta. He tratado de adaptar y desarrollar una solución que se adapte a nuestro contexto, para que las personas Guatemaltecas puedan vender, comprar e intercambiar productos con la facilidad de un click. Analicemos el sistema basándonos en la estructura CEEM:

### Elementos:

- **Producto:** es el elemento fundamental del sistema, sin él no tendría una razón para existir el sistema. Ya que lo que buscamos es vender, comprar o intercambiar productos.
- **Usuario:** es otro elemento fundamental para el sistema, ya que ellos crearán posts, ofertas y completarán el objetivo del sistema. Así mismo, permiten tener un control al ser miembros de soporte y administradores.
- **Plataforma Web:** en este caso, es uno de los medios donde se desarrolla nuestro sistema, y es el medio que utilizan nuestros usuarios para completar el objetivo del mismo.
- **Empresas de Envío:** ellos permitirán la entrega de los productos, al recibir las órdenes de envío desde nuestro sistema, el sistema que utilizan ellos es un caja negra ya que no sabemos como funciona, solo conocemos las entradas y salidas del mismo.
- **Posts y ofertas:** estos elementos permiten que se desarrolle el objetivo del sistema, son parte de las entradas al sistema que generan una salida (envíos).

### **Medio:**

- **Plataforma Web:** se refiere al sistema que hemos diseñado, aplicando diferentes tecnologías web para una cómoda y simple utilización del sistema.
- **Infraestructura Informática:** garantiza alta disponibilidad para la plataforma web, asegurando que la plataforma pueda llegar a todas partes y funcione correctamente.
- **Empresas de Envío:** permiten que los usuarios obtengan sus productos y se completen las ventas, compras e intercambios.
- **Métodos de Pago:** permiten que se completen las compras, también es una caja negra ya que solo sabemos el tipo de pago, sin embargo, no sabemos cómo.

### **Entradas:**

- **Producto:** se publican para su venta, para la cual se deben hacer ofertas. Así mismo se deben crear detalles para los productos, de esta manera podemos definir parámetros como el precio, descripción, etc.

### **Salidas:**

- **Productos vendidos, comprados o intercambiados:** son productos que han sido procesados por medio de las publicaciones y ofertas, que generan envíos para su entrega. Como medio final, salen del sistema como artículos vendidos, comprados o intercambiados.

**Relaciones:**

- Creación.
- Compra.
- Venta.
- Intercambio.
- Envío.
- Pago.
- Negociación.
- Acuerdos.

**Objetivo:**

- Facilitar las compras, ventas e intercambios de productos para la población Guatemalteca, con una solución al alcance de todos, mediante una plataforma web accesible desde computadoras y dispositivos móviles. Con funcionalidades, como gestión de inventario, creación de usuarios, publicación de productos, publicación de ofertas, gestión de compras. ventas e intercambios, y envíos a todo el país.

**Actividad:**

- Compra, venta e intercambio de productos de la población guatemalteca.

**Cajas negras:**

- Sistema de Envíos.
- Sistema de pagos.
- Infraestructura Informática.

**Cajas blancas:**

- Sistema de Creación de usuarios.
- Sistema de Creación de Publicaciones y Ofertas.
- Página principal que muestra los productos disponibles.
- Sistema de Inventario de productos por usuario.
- Sistema de generación de órdenes de envío.

## PRINCIPALES OBJETIVOS

Para asegurar que los principales usuarios de la plataforma experimenten comodidad y eficiencia al utilizarla, se han establecido una serie de objetivos clave en el desarrollo del sistema:

- **Estructuración de la Información:** Uno de los objetivos fundamentales es garantizar una organización clara y coherente de la información en la plataforma. Esto se logrará mediante una estructura de navegación lógica y una categorización adecuada del contenido. La disposición ordenada de los elementos permitirá a los usuarios localizar fácilmente la información que buscan, mejorando así la experiencia del usuario.
- **Diseño Responsivo:** El diseño de la plataforma se adaptará de manera fluida a diferentes tamaños de pantalla y dispositivos, asegurando que los usuarios puedan acceder y utilizar la página de manera efectiva tanto en computadoras de escritorio como en tablets y dispositivos móviles. Un diseño responsivo es esencial para proporcionar una experiencia consistente independientemente del dispositivo utilizado.
- **Pruebas de Usabilidad:** Se realizarán pruebas exhaustivas de usabilidad con un grupo representativo de usuarios. Estas pruebas ayudarán a identificar posibles obstáculos en la navegación, comprensión de la interfaz y flujo de interacción. Los resultados de estas pruebas guiarán las mejoras necesarias para optimizar la experiencia del usuario y corregir posibles problemas.
- **Diseño Intuitivo:** El diseño de la plataforma se orientará hacia la intuición y la facilidad de uso. Se evitarán elementos confusos o complicados que puedan dificultar la interacción. La disposición de botones, menús y elementos interactivos se basará en patrones familiares que los usuarios puedan comprender de manera intuitiva, reduciendo la curva de aprendizaje y fomentando una experiencia fluida desde el primer momento.

Cumpliendo estos objetivos, se garantizará que la plataforma no solo satisfaga las necesidades y expectativas de los usuarios, sino que también les brinde una experiencia agradable y satisfactoria.

## **MARCO TEÓRICO**

### **Capítulo 1.**

#### **1.1 Servidor HTTP Apache.**

Apache HTTP Server, comúnmente conocido como Apache, es un servidor web de código abierto ampliamente utilizado en Internet. Es un software que recibe solicitudes de clientes web, como navegadores, y les envía las páginas web correspondientes.

Apache es una de las opciones más populares para alojar sitios web debido a su estabilidad, flexibilidad y capacidad de personalización. Es compatible con una amplia variedad de sistemas operativos y se integra bien con otros componentes de la pila LAMP (Linux, Apache, MySQL, PHP), que es esencial en el desarrollo web.

#### **1.2 MySQL.**

MySQL es un sistema de gestión de bases de datos (DBMS) de código abierto que permite almacenar, organizar y recuperar datos. Se utiliza principalmente para aplicaciones web y es compatible con múltiples lenguajes de programación.

MySQL es esencial para almacenar y gestionar datos en aplicaciones web. Se utiliza en combinación con PHP y un sinnúmero de lenguajes de programación para crear aplicaciones dinámicas que interactúan con bases de datos. Usa el lenguaje SQL diseñado para gestionar y manipular bases de datos relacionales. Fue desarrollado en la década de 1970 y se ha convertido en un estándar en la gestión de bases de datos. SQL se utiliza para realizar una variedad de tareas relacionadas con datos, como la creación, modificación y consulta de bases de datos.

#### **1.3 PHP.**

PHP (Hypertext Preprocessor) es un lenguaje de programación de código abierto diseñado para el desarrollo web. Se utiliza para crear aplicaciones web dinámicas y se puede integrar fácilmente con HTML. PHP es un lenguaje esencial para la creación de sitios web dinámicos, ya que permite a los desarrolladores interactuar con bases de datos, procesar formularios y generar contenido web en tiempo real.

## **1.4 HTML.**

HTML (Hypertext Markup Language) es el lenguaje de marcado utilizado para crear páginas web. Define la estructura y el contenido de una página web mediante etiquetas. HTML es el lenguaje base de la web y es esencial para la creación de páginas web. Define la estructura de una página y cómo se muestran los elementos en un navegador.

## **1.5 JavaScript.**

JavaScript es un lenguaje de programación utilizado para agregar interactividad y funcionalidad a las páginas web. Se ejecuta en el navegador del usuario. JavaScript es fundamental para crear sitios web interactivos y dinámicos. Permite realizar validaciones de formularios, animaciones, efectos visuales y más.

## **1.6 CSS.**

CSS (Cascading Style Sheets) es un lenguaje utilizado para definir el estilo y la presentación de una página web. Permite controlar la apariencia de los elementos HTML. CSS es esencial para el diseño web, ya que permite dar formato y estilo a los elementos HTML, lo que mejora la apariencia y la usabilidad de un sitio web.

## **1.7 Xampp.**

XAMPP es un paquete de software que incluye Apache, MySQL, PHP y Perl, diseñado para facilitar la configuración de un entorno de desarrollo web local en sistemas Windows, macOS y Linux. XAMPP es una herramienta valiosa para desarrolladores web, ya que permite crear y probar sitios web en un entorno de desarrollo local antes de desplegarlos en un servidor en vivo.

## **1.8 PhpMyAdmin.**

PhpMyAdmin es una herramienta de administración de bases de datos web que proporciona una interfaz gráfica para administrar bases de datos MySQL a través de un navegador web. PhpMyAdmin facilita la gestión y administración de bases de datos MySQL, lo que es crucial para los desarrolladores web que trabajan con aplicaciones que utilizan MySQL como motor de base de datos.



## **1.9 Visual Studio Code.**

Visual Studio Code (VS Code) es un editor de código fuente de código abierto desarrollado por Microsoft. Es ampliamente utilizado en el desarrollo web y ofrece numerosas extensiones para admitir diferentes lenguajes de programación y herramientas. VS Code es un entorno de desarrollo altamente personalizable y eficiente que facilita la escritura, depuración y gestión de código para proyectos web y aplicaciones.

## Capítulo 2.

### 2.2 Backend.

Es la parte "oculta" de una aplicación web que se encuentra en el servidor. Realiza tareas que no son visibles para el usuario, como procesar solicitudes, administrar bases de datos, autenticar usuarios y gestionar la lógica empresarial. Los elementos clave del Backend son:

- **Servidor Web:** El servidor web, como Apache o Nginx, maneja las solicitudes entrantes desde el frontend y envía respuestas adecuadas, como páginas HTML, datos JSON o imágenes.
- **Lenguajes de Programación del Backend:** Los lenguajes de programación comunes para el backend incluyen PHP, Python, Ruby, Node.js (JavaScript en el lado del servidor) y más. Estos lenguajes se utilizan para crear la lógica de la aplicación y gestionar las operaciones del servidor.
- **Base de Datos:** El backend suele interactuar con una base de datos para almacenar y recuperar datos. Los sistemas de gestión de bases de datos, como MySQL, PostgreSQL o MongoDB, son comunes en el backend.
- **Autenticación y Seguridad:** El backend se encarga de la autenticación de usuarios, la autorización y la seguridad, asegurando que los datos y las operaciones estén protegidos.
- **API (Interfaz de Programación de Aplicaciones):** El backend a menudo expone API que permiten a aplicaciones frontend o terceros acceder a datos y funcionalidades de la aplicación de forma controlada.

En resumen, el Backend se ocupa de las tareas detrás de escena que hacen que una aplicación web funcione de manera efectiva. Ambos componentes son esenciales y trabajan juntos para proporcionar una experiencia completa en una aplicación web.

## 2.3 Frontend.

Es la parte de una aplicación web que los usuarios ven y con la que interactúan directamente en sus navegadores web. Incluye la interfaz de usuario, el diseño, la presentación y la experiencia del usuario. Los elementos clave del frontend son:

- **HTML (Hypertext Markup Language):** HTML es el lenguaje de marcado utilizado para estructurar y crear la parte visual de una página web. Define la disposición de los elementos en una página, como encabezados, párrafos, imágenes, formularios, enlaces y más.
- **CSS (Cascading Style Sheets):** CSS se utiliza para dar estilo y formato a los elementos HTML. Define la apariencia de la página, incluidos aspectos como colores, fuentes, márgenes, espaciado y diseño general.
- **JavaScript:** JavaScript es un lenguaje de programación que agrega interactividad y funcionalidad dinámica a una página web. Permite crear efectos visuales, realizar validaciones de formularios, cargar datos de forma asincrónica y mucho más.
- **Frameworks y Bibliotecas:** Los desarrolladores Frontend a menudo utilizan Frameworks y bibliotecas como React, Angular o Vue.js para simplificar y agilizar el desarrollo de aplicaciones web interactivas y ricas en contenido.

El Frontend es lo que los usuarios ven y con lo que interactúan directamente al visitar un sitio web. Su objetivo principal es proporcionar una experiencia de usuario atractiva y funcional.

## **Capítulo 3.**

### **3.1 Modelo Vista-Controlador (MVC).**

El Modelo-Vista-Controlador (MVC) es un patrón de diseño arquitectónico ampliamente utilizado en el desarrollo web y de aplicaciones para separar de manera eficiente las diferentes responsabilidades y componentes de una aplicación. Este patrón ayuda a organizar el código de una manera que mejora la modularidad, la reutilización y la mantenibilidad de la aplicación. A continuación, se define el MVC en el contexto del desarrollo web:

#### **1. Modelo (Model):**

El Modelo representa la capa de datos y la lógica de negocio de la aplicación. En esta capa, se gestionan los datos, su almacenamiento, recuperación y procesamiento. Los modelos suelen comunicarse con la base de datos o cualquier otro almacenamiento de datos para obtener o actualizar información. No deben tener conocimiento de la interfaz de usuario ni de la lógica de presentación. Son independientes de cómo se muestran los datos.

#### **2. Vista (View):**

La Vista es la capa encargada de la presentación de los datos al usuario. Define cómo se muestran los datos al cliente. Las vistas son pasivas y reflejan el estado actual del Modelo. No realizan operaciones de lógica de negocio ni manipulan directamente los datos. Pueden existir múltiples vistas para un mismo modelo, lo que permite mostrar la misma información de diferentes maneras.

### 3. Controlador (Controller):

El Controlador actúa como intermediario entre el Modelo y la Vista. Recibe las solicitudes del usuario a través de la interfaz de usuario y gestiona las acciones correspondientes. Se encarga de la lógica de negocio, toma decisiones basadas en la entrada del usuario y actualiza el Modelo en consecuencia. También actualiza la Vista para reflejar los cambios en el Modelo, lo que permite que los datos se presenten de manera coherente y actualizada.

En resumen, el patrón MVC divide una aplicación web en tres componentes principales, cada uno con una responsabilidad claramente definida:

- El Modelo se encarga de la gestión de datos y la lógica de negocio.
- La Vista se encarga de la presentación y la interfaz de usuario.
- El Controlador se encarga de coordinar las acciones del usuario, la lógica de negocio y la actualización de la Vista y el Modelo.

Esta separación de preocupaciones facilita el desarrollo colaborativo, la escalabilidad y la mantenibilidad de las aplicaciones web, ya que cada componente puede ser desarrollado, probado y modificado de forma independiente.

## **VERSIONES DE SOFTWARE**

**Xampp Control Panel:** v3.30

### **Servidor de base de datos:**

- Servidor: 127.0.0.1 via TCP/IP
- Tipo de servidor: MariaDB
- Conexión del servidor: No se está utilizando SSL Documentación
- Versión del servidor: 10.4.32-MariaDB - mariadb.org binary distribution
- Versión del protocolo: 10
- Usuario: root@localhost
- Conjunto de caracteres del servidor: UTF-8 Unicode (utf8mb4)

### **Servidor web:**

- Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
- Versión del cliente de base de datos: libmysql - mysqlnd 8.2.12
- extensión PHP: mysqli Documentación curl Documentación mbstring Documentación
- Versión de PHP: 8.2.12

**phpMyAdmin:** version 5.2.1

# DIAGRAMA ENTIDAD RELACIÓN



