

*Universidad de San Carlos de Guatemala*  
*Centro Universitario de Occidente*  
*División de Ciencias de la Ingeniería*  
*Ingeniería en Ciencias y Sistemas.*  
*Teoría de Sistemas 1*



**TradeChatApp**

**Enlace Repositorio**

**<https://github.com/MunguiaMander/TradeChatApp>**

**Manual Técnico**

**Práctica “Hola Mundo” Laboratorio de Teoría de Sistemas 1.**

**Septiembre 5 del 2,024**

**MARCO JOSE MUNGUIA ALVA - 201931804**

# Índice

<b>Índice</b>	<b>1</b>
<b>Tecnologías Utilizadas en el Sistema Informático</b>	<b>2</b>
No Instalables:	2
HTML (HyperText Markup Language) ver 5.0:	2
CSS (Cascade Style Sheets) ver 3.0:	2
JavaScript (JS) ver ES2023:	2
Instalables:	2
MySQL ver (8.1.0):	2
XAMPP ver (8.2.7):	2
PHP (Hypertext Preprocessor) ver (8.3.11) thread support:	3
<b>Iniciar TradeChatApp Windows10-11:</b>	<b>4</b>
Como iniciar la configuración del proyecto:	4
<b>Acerca de la Programación</b>	<b>6</b>
Sobre el patrón MVC:	6
Estructura del Proyecto:	6

# Tecnologías Utilizadas en el Sistema Informático

## No Instalables:

### HTML (HyperText Markup Language) *ver 5.0:*

HTML es el lenguaje estándar para la creación de páginas web. Proporciona la estructura básica de una página web, permitiendo la integración de texto, imágenes, y enlaces.

Instalación:

No requiere instalación específica. Solo necesitas un editor de texto (como Visual Studio Code, Sublime Text, o Notepad++) para escribir archivos `.html` y un navegador web para abrirlos.

### CSS (Cascade Style Sheets) *ver 3.0:*

CSS es el lenguaje de estilos utilizado para describir la presentación de documentos HTML. Permite definir colores, fuentes, diseño y la apariencia visual de las páginas web.

Instalación:

Como HTML, CSS no requiere instalación específica. Puedes crear archivos `.css` en cualquier editor de texto y enlazarlos a documentos HTML. Los navegadores web interpretan CSS.

### JavaScript (JS) *ver ES2023:*

JavaScript es un lenguaje de programación utilizado principalmente en el desarrollo web para crear contenido interactivo en páginas web. Permite implementar funciones dinámicas, como validación de formularios, animaciones, y comunicación con servidores.

Instalación:

No requiere instalación para su uso básico en navegadores, ya que todos los navegadores modernos tienen un motor de JavaScript integrado.

## Instalables:

### MySQL *ver (8.1.0):*

MySQL es un sistema de gestión de bases de datos relacional (RDBMS) que utiliza SQL (Structured Query Language) para gestionar y manipular bases de datos. Es uno de los sistemas de bases de datos más populares del mundo, especialmente para aplicaciones web.

Instalación:

1. Descarga MySQL desde su [sitio oficial](#).
2. Sigue el instalador guiado para configurar el servidor MySQL.
3. Durante la instalación, puedes elegir las opciones por defecto y establecer una contraseña para el usuario "root".
4. Utiliza MySQL Workbench o el terminal para gestionar las bases de datos.

### XAMPP *ver (8.2.7):*

XAMPP es una distribución de Apache fácil de instalar que incluye servidores Apache, MySQL, PHP y Perl. Es ideal para desarrollar localmente aplicaciones web en PHP y MySQL sin necesidad de configurar estos componentes por separado.

Instalación:

1. Descarga XAMPP desde su [sitio oficial](#).
2. Ejecuta el instalador y selecciona los componentes que desees (generalmente Apache, MySQL, PHP).

3. Una vez instalado, inicia el "Panel de Control de XAMPP" y activa Apache y MySQL para ejecutar un servidor web local.
4. Coloca tus archivos en la carpeta `htdocs` para que puedan ser accedidos desde el navegador usando `http://localhost/`.

**PHP (Hypertext Preprocessor) ver (8.3.11) thread support:**

PHP es un lenguaje de scripting del lado del servidor diseñado para el desarrollo web. Se integra con HTML y es capaz de interactuar con bases de datos, realizar operaciones en el servidor, y generar contenido dinámico en páginas web.

**Instalación:**

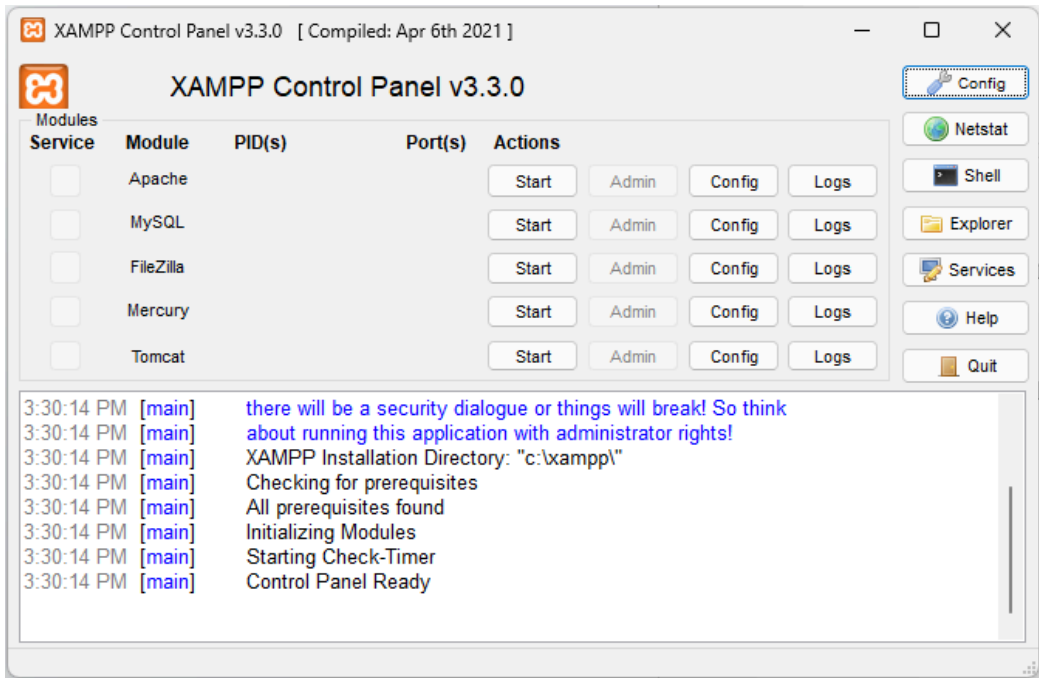
Si estás usando XAMPP, PHP ya viene preinstalado. Sin embargo, si prefieres instalar PHP por separado:

1. Descarga PHP desde su [sitio oficial](#).
2. Extrae el archivo descargado en una carpeta de tu elección.
3. Configura la variable de entorno del sistema para que Windows reconozca el comando `php`.
4. Desde la línea de comandos, ejecuta `php -v` para verificar la instalación.

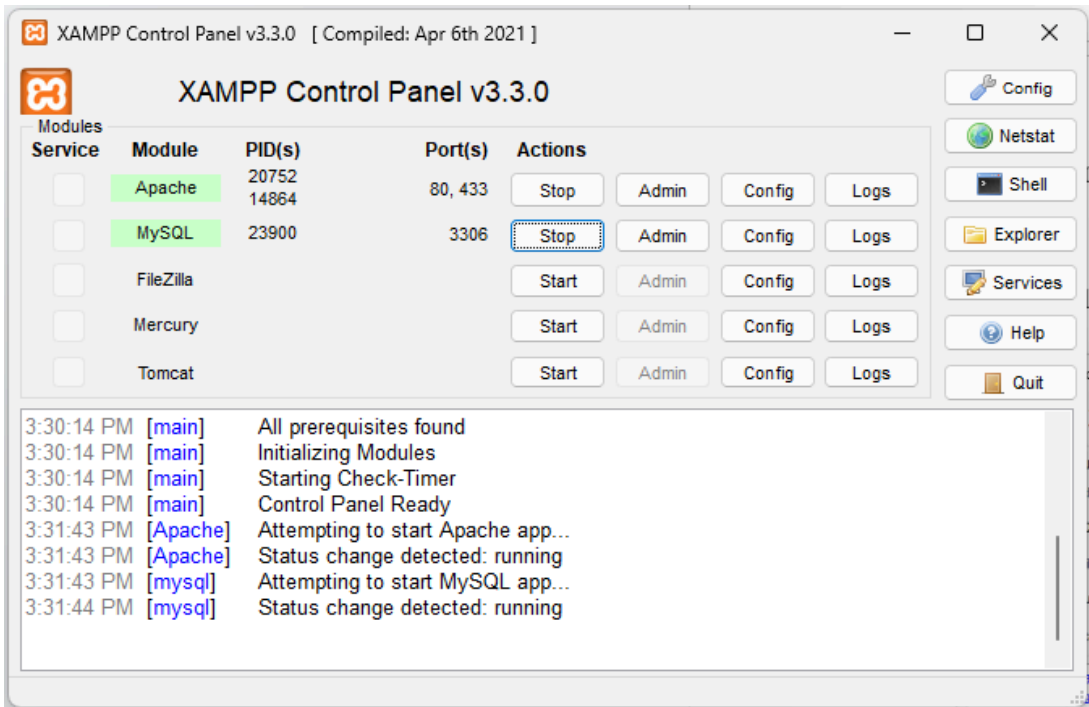
# Iniciar TradeChatApp Windows10-11:

## Como iniciar la configuración del proyecto:

Previamente instalados las tecnologías necesarias para poder iniciar el proyecto, es momento de iniciar este en una máquina (computador personal, servidor en la nube, máquina virtual, etc...). Iniciamos XAMPP:

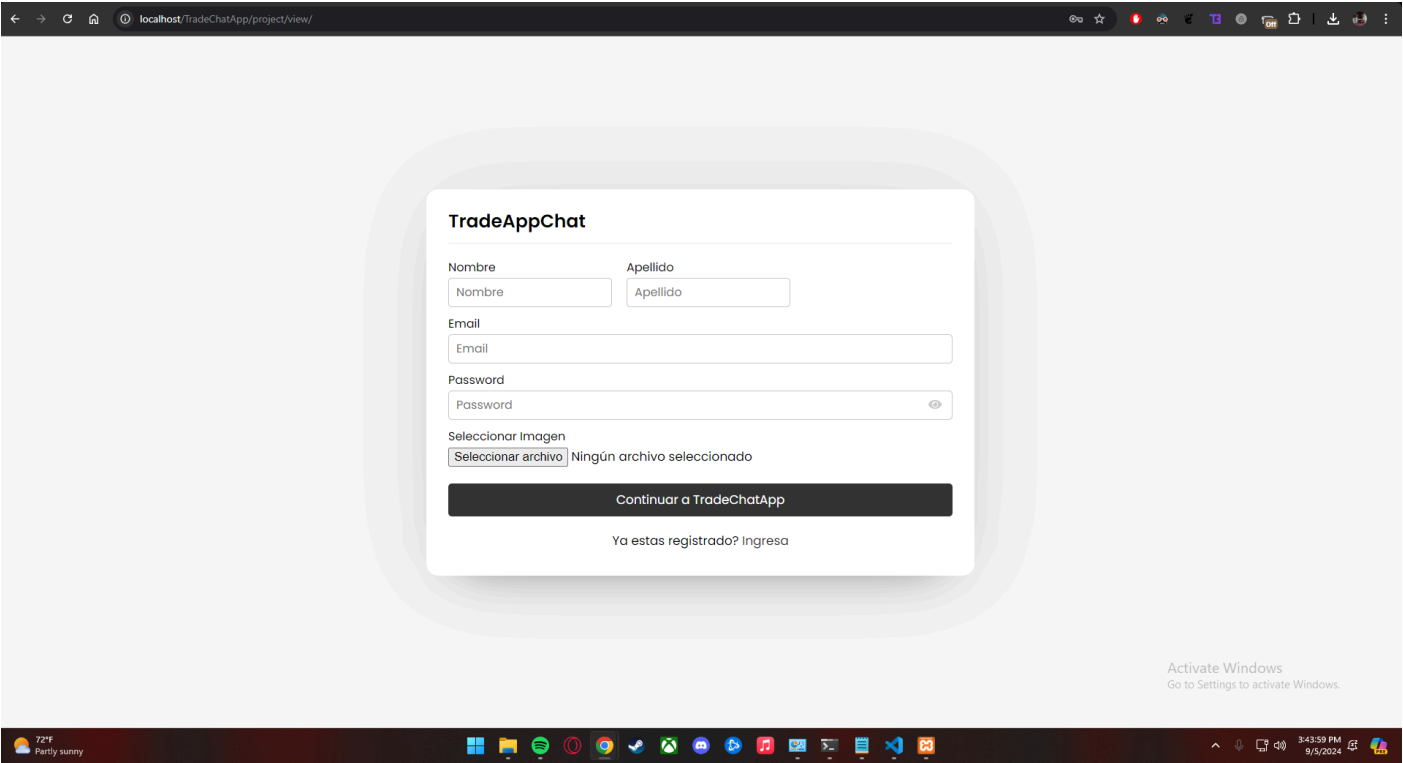


Procedemos a iniciar el *Module* de Apache presionando el boton **Start** en la columna de Actions y de la misma manera tambien iniciamos el *Module* de MySQL:



Es importante que Apache esté en el **Port(s): 80 e 433** y MySQL en el **Port: 3306**. Esto para que puedas seguir las rutas que se indicaran en la documentación. De no ser asi puedes configurar los puertos en los archivos de configuración **Apache (httpd.conf, httpd-ssl.conf)** y para MySQL en **MySQL (my.ini)**.

Mueve la carpeta del proyecto *TradeChatApp* a la ruta **C:\xampp\htdocs** de esta manera ya podrás acceder a la ruta **https://localhost:80/TradeChatApp/ptoject/view/index.php** y tendrás una vista principal del proyecto la cual enseñaremos a cómo utilizar en el *Manual de Usuario*.

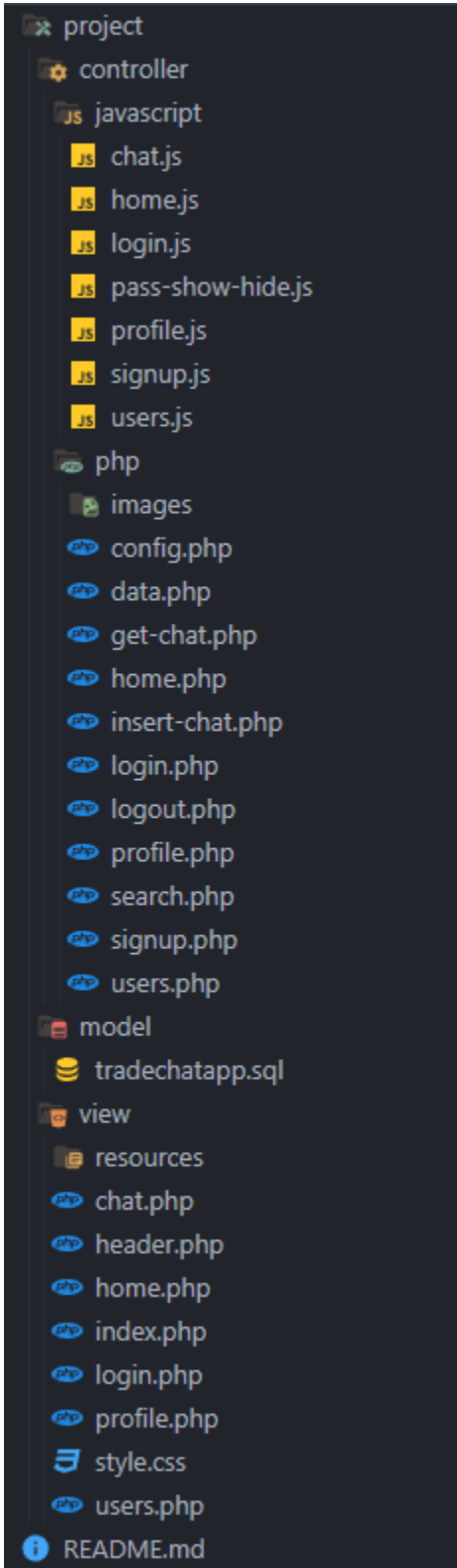


## Acerca de la Programación

### Sobre el patrón MVC:

El patrón MVC es una arquitectura de software utilizada para separar las preocupaciones de una aplicación en tres componentes principales: Modelo, Vista y Controlador. Este patrón fue implementado en el proyecto utilizando las tecnologías HTML, CSS, JavaScript, PHP, MySQL y XAMPP.

### Estructura del Proyecto:



Este es el arbol de trabajo de archivos del proyecto. Este arbol esta creado siguiendo las reglas de negocio aplicada a la practica previamente descritas en el [Marco Teorico](#), por ende;

- En el directorio **model** encontraremos el script de la base de datos la cual contiene las tablas que son necesarias para el funcionamiento del modulo descrito en el marco teorico. Siendo las más importantes la de **messages** y la de **users** ya que sobre estas gira el funcionamiento principal del chat de contacto entre compradores y vendedores. Notese que esta almacena mensajes de entrada y salida por usuario esto para que el controlador pueda recuperar las conversaciones con cada inicio de sesion asi como el nombre de las img que seran cargadas a cada perfil.

```
-- Database: `tradechatapp`
```

```

CREATE TABLE IF NOT EXISTS `messages` (
  `msg_id` int(11) NOT NULL AUTO_INCREMENT,
  `incoming_msg_id` int(255) NOT NULL,
  `outgoing_msg_id` int(255) NOT NULL,
  `msg` varchar(1000) NOT NULL,
  PRIMARY KEY (`msg_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE IF NOT EXISTS `users` (
  `user_id` int(11) NOT NULL AUTO_INCREMENT,
  `unique_id` int(255) NOT NULL,
  `fname` varchar(255) NOT NULL,
  `lname` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `img` varchar(255) NOT NULL,
  `status` varchar(255) NOT NULL,
  PRIMARY KEY (`user_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

### ***model/tradechatapp.sql***

- messages: Almacena los mensajes enviados entre usuarios, incluyendo los identificadores del emisor y receptor, y el contenido del mensaje.
  - users: Almacena la información de los usuarios, incluyendo su nombre, correo electrónico, contraseña, imagen de perfil y estado actual.
  - post: Almacena publicaciones creadas por los usuarios, registrando el contenido de la publicación y la fecha de creación.
  - article: Almacena artículos escritos por los usuarios, con un título, contenido y la fecha de publicación.
  - pay: Almacena los detalles de pagos o transacciones de los usuarios, como el monto, método de pago y la fecha de la transacción.
  - report: Almacena reportes de usuarios, registrando quién reporta, quién es reportado, la razón y los detalles del reporte.
- En el directorio de ***controller*** encontraremos las clases que son necesarias para la comunicación entre el cliente desde las vistas de pagina con el sistema, ya sea base de datos para algunos casos, o solo otras vistas.

### ***controller/php***

- config.php: Archivo de configuración para la conexión a la base de datos y otros parámetros de configuración global.
- data.php: Gestiona la obtención o manipulación de datos en el sistema, probablemente relacionado con la base de datos.
- get-chat.php: Obtiene los mensajes de chat entre los usuarios desde la base de datos.
- home.php: Controla la lógica de la página de inicio o el dashboard del usuario después de iniciar sesión.
- insert-chat.php: Se encarga de insertar nuevos mensajes en la base de datos para el sistema de chat.
- login.php: Maneja la lógica del inicio de sesión del usuario, incluyendo validaciones y autenticación con la base de datos.
- logout.php: Cierra la sesión del usuario y redirige a la página de inicio de sesión.
- profile.php: Controla la lógica para mostrar y actualizar el perfil del usuario.



- `search.php`: Permite la búsqueda de usuarios o mensajes dentro del sistema.
- `signup.php`: Gestiona el registro de nuevos usuarios, guardando la información en la base de datos.
- `users.php`: Muestra una lista de usuarios registrados o gestionados, posiblemente incluyendo filtros o listados interactivos.

#### *controller/javascript*

- `home.js`: Controla las interacciones dinámicas en la página de inicio, posiblemente gestionando eventos o actualizaciones en tiempo real.
  - `login.js`: Maneja la lógica de validación y envío de datos del formulario de inicio de sesión mediante AJAX.
  - `pass-show-hide.js`: Añade la funcionalidad de mostrar/ocultar la contraseña en los campos de formulario.
  - `signup.js`: Gestiona la validación y envío de datos del formulario de registro mediante AJAX.
  - `users.js`: Controla la lógica para mostrar y gestionar la lista de usuarios, incluyendo posibles interacciones como la búsqueda.
- En el directorio ***view*** encontraremos todo lo relacionado a la vistas con la que interactúa el usuario.
    - `chat.php`: Vista que presenta la interfaz de chat para los usuarios, mostrando mensajes y permitiendo la interacción en tiempo real.
    - `header.php`: Archivo reutilizable para la cabecera, normalmente incluye navegación y estilos comunes en todas las páginas.
    - `home.php`: La interfaz principal o dashboard que ve el usuario después de iniciar sesión.
    - `index.php`: Página de inicio del sistema, probablemente incluye formularios para login o registro.
    - `login.php`: Formulario para que los usuarios inicien sesión en el sistema.
    - `profile.php`: Página que muestra los detalles del perfil del usuario y permite la edición de información personal.
    - `style.css`: Archivo de estilos que define el diseño visual de todas las vistas.
    - `users.php`: Vista que muestra la lista de usuarios disponibles en el sistema, probablemente con opciones de búsqueda o filtrado.