

2024

MANUAL INSTALACIÓN

Sistema de control y registro de encomiendas



Manual de Instalación

Link de Git: https://github.com/Odaliz21/TromeExpress_Tl.git

Instalación de XAMPP

1. Dirígete a la página oficial de XAMPP: <https://www.apachefriends.org/download.html>
2. Selecciona la versión compatible con tu sistema operativo (Windows, Linux o MacOS).
3. Descarga el archivo ejecutable.
4. Abre el instalador y sigue las instrucciones en pantalla para completar la instalación.
5. Una vez instalado, abre el Panel de Control de XAMPP y asegúrate de iniciar los servicios de Apache y MySQL.

Instalación de MySQL Workbench

1. Dirígete a la página oficial de MySQL: <https://dev.mysql.com/downloads/workbench/>
2. Descarga MySQL Workbench seleccionando tu sistema operativo.
3. Abre el archivo de instalación y sigue los pasos en pantalla.
4. Durante la instalación, asegúrate de que el servidor MySQL esté correctamente configurado.

Instalación de Visual Studio Code

1. Visita el sitio oficial de Visual Studio Code: <https://code.visualstudio.com/>
2. Descarga la versión adecuada para tu sistema operativo.
3. Instala el programa siguiendo las instrucciones en pantalla.
4. Una vez instalado, abre VSCode y añade las extensiones necesarias para trabajar con JavaScript, Node.js y React.

Instalación de Node.js

1. Ve a la página oficial de Node.js: <https://nodejs.org/en/download/>

2. Descarga la versión LTS (Long Term Support) adecuada para tu sistema operativo.
3. Abre el archivo descargado e instálalo siguiendo las indicaciones del instalador.
4. Una vez instalado, abre una terminal o consola y escribe el comando ``node -v`` para verificar que Node.js está correctamente instalado.
5. Instala también npm (Node Package Manager), que viene incluido con Node.js, escribiendo ``npm -v`` en la consola.

Instalación de React

1. Abre la terminal o consola de comandos en tu sistema operativo.
2. Asegúrate de tener Node.js instalado (puedes verificarlo con ``node -v``).
3. Para crear una nueva aplicación en React, escribe el siguiente comando: ``npx create-react-app my-app``
4. Una vez finalizada la instalación, navega a tu nuevo proyecto con el comando: ``cd my-app``
5. Inicia la aplicación ejecutando: ``npm start``

Dependencias necesarias para el Backend en Node.js

Para construir el backend de la aplicación utilizando Node.js y Express, asegúrate de instalar las siguientes dependencias:

1. Express:

- Framework web para Node.js que facilita la creación de aplicaciones.

- Instalación:

```
```bash
npm install express
```
```

2. MySQL2:

- Cliente MySQL para Node.js que permite interactuar con una base de datos MySQL.

- Instalación:

```
```bash
```

```
npm install mysql2
```

### 3. Dotenv:

- Permite cargar variables de entorno desde un archivo `.env`.

- Instalación:

```
```bash
```

```
npm install dotenv
```

4. Bcryptjs:

- Librería para encriptar contraseñas de manera segura.

- Instalación:

```
```bash
```

```
npm install bcryptjs
```

```
```
```

5. JsonWebToken:

- Librería para crear y verificar JSON Web Tokens (JWT), útil para la autenticación.

- Instalación:

```
```bash
```

```
npm install jsonwebtoken
```

```
```
```

6. Body-Parser (opcional):

- Middleware para analizar los cuerpos de las solicitudes (aunque Express ya incluye un middleware para esto).

- Instalación:

```
```bash
```

```
npm install body-parser
```

```
...
```

## 7. Nodemon (solo para desarrollo):

- Herramienta que reinicia automáticamente el servidor cuando se detectan cambios en los archivos.

- Instalación:

```
```bash
```

```
npm install --save-dev nodemon
```

Comando para Instalar Todas las Dependencias

Puedes instalar todas estas dependencias a la vez ejecutando el siguiente comando en la terminal de tu proyecto:

```
```bash
```

```
npm install express mysql2 dotenv bcryptjs jsonwebtoken body-parser
```

```
npm install --save-dev nodemon
```

## Pruebas para la API de Clientes y Oficinas

### ## Pruebas para Clientes

#### ### 1. Crear Cliente

- **Método**: POST

- **URL**: http://localhost:3000/api/clientes

- **Cuerpo (JSON)**:

```
{
```

```
 "codCliente": "C001",
```

```
"Nombre": "Juan",
"Apellido": "Pérez",
"DNI": "12345678",
"Tipo": "Remitente"
}
- **Respuesta Esperada**:

{
 "message": "Cliente creado exitosamente"
}
```

### ### 2. Obtener Todos los Clientes

```
- **Método**: GET
- **URL**: http://localhost:3000/api/clientes
- **Respuesta Esperada**:

[
 {
 "codCliente": "C001",
 "Nombre": "Juan",
 "Apellido": "Pérez",
 "DNI": "12345678",
 "Tipo": "Remitente"
 }
]
```

### ### 3. Obtener Cliente por ID

```
- **Método**: GET
```

- **\*\*URL\*\***: http://localhost:3000/api/clientes/C001

- **\*\*Respuesta Esperada\*\***:

```
{
 "codCliente": "C001",
 "Nombre": "Juan",
 "Apellido": "Pérez",
 "DNI": "12345678",
 "Tipo": "Remitente"
}
```

#### ### 4. Actualizar Cliente

- **\*\*Método\*\***: PUT

- **\*\*URL\*\***: http://localhost:3000/api/clientes/C001

- **\*\*Cuerpo (JSON)\*\***:

```
{
 "Nombre": "Juan Carlos",
 "Apellido": "Pérez",
 "DNI": "12345678",
 "Tipo": "Remitente"
}
```

- **\*\*Respuesta Esperada\*\***:

```
{
 "message": "Cliente actualizado exitosamente"
}
```

#### ### 5. Eliminar Cliente

- **Método**: DELETE
- **URL**: http://localhost:3000/api/clientes/C001
- **Respuesta Esperada**:  

```
{
 "message": "Cliente eliminado exitosamente"
}
```

## ## Pruebas para Oficinas

### ### 1. Crear Oficina

- **Método**: POST
- **URL**: http://localhost:3000/api/oficinas
- **Cuerpo (JSON)**:  

```
{
 "codOficina": "0001",
 "NombreOficina": "Cusco",
 "Direccion": "Av. Republica de Argentina",
 "Ciudad": "Cusco",
 "Telefono": "999999999"
}
```
- **Respuesta Esperada**:  

```
{
 "message": "Oficina creada exitosamente"
}
```



### ### 2. Obtener Todas las Oficinas

- **Método**: GET

- **URL**: http://localhost:3000/api/oficinas

- **Respuesta Esperada**:

```
[
 {
 "codOficina": "0001",
 "NombreOficina": "Cusco",
 "Direccion": "Av. Republica de Argentina",
 "Ciudad": "Cusco",
 "Telefono": "999999999"
 }
]
```

### ### 3. Obtener Oficina por ID

- **Método**: GET

- **URL**: http://localhost:3000/api/oficinas/0001

- **Respuesta Esperada**:

```
{
 "codOficina": "0001",
 "NombreOficina": "Cusco",
 "Direccion": "Av. Republica de Argentina",
 "Ciudad": "Cusco",
 "Telefono": "999999999"
}
```

#### 4. Actualizar Oficina

- **Método**: PUT

- **URL**: http://localhost:3000/api/oficinas/0001

- **Cuerpo (JSON)**:

```
{
 "NombreOficina": "Oficina Central Cusco",
 "Direccion": "Av. Republica de Argentina",
 "Ciudad": "Cusco",
 "Telefono": "8888888888"
}
```

- **Respuesta Esperada**:

```
{
 "message": "Oficina actualizada exitosamente"
}
```

#### ### 5. Eliminar Oficina

- **Método**: DELETE

- **URL**: http://localhost:3000/api/oficinas/0001

- **Respuesta Esperada**:

```
{
 "message": "Oficina eliminada exitosamente"
}
```