

# Documentación Completa: Training-Didactic-PLC-WIFI

## Descripción del Proyecto

Este repositorio contiene los archivos fuente del proyecto "Training Didactic PLC-WIFI", incluyendo código basado en Python para una interfaz web llamada "HMI Web", así como código MicroPython desarrollado para el microcontrolador ESP32.

## ¿Qué aprenderás con este proyecto?

- Programación de PLCs (Controladores Lógicos Programables) de forma didáctica
- Uso de ESP32 con conectividad WiFi
- Desarrollo de interfaces HMI (Human Machine Interface) web
- Programación en MicroPython
- Comunicación entre dispositivos IoT

## Audiencia Objetivo

Este proyecto está diseñado para:

- Estudiantes de ingeniería industrial, electrónica o mecatrónica
- Profesionales que quieren aprender sobre automatización industrial
- Personas interesadas en proyectos IoT con ESP32
- Cualquier persona sin experiencia previa en Git que quiera aprender

## Requisitos del Sistema

### Hardware Necesario

- **ESP32:** Microcontrolador principal del proyecto
- **Computadora:** Windows, Mac o Linux
- **Cable USB:** Para programar el ESP32
- **Conexión a Internet:** Para descargar herramientas y librerías

### Software que Instalaremos

- **Git:** Para descargar y gestionar el proyecto
- **Python 3.7+:** Para ejecutar la interfaz web
- **Thonny IDE:** Para programar el ESP32 con MicroPython

- **Navegador web:** Para usar la interfaz HMI

## **Conceptos Básicos que Debes Conocer**

### **¿Qué es un PLC?**

Un PLC (Programmable Logic Controller) es un dispositivo electrónico robusto usado en la industria para:

- Controlar máquinas automáticamente
- Leer sensores (temperatura, presión, movimiento)
- Activar actuadores (motores, válvulas, luces)
- Ejecutar lógica de control programada

### **¿Qué es una HMI?**

Una HMI (Human Machine Interface) es una interfaz gráfica que permite:

- Visualizar el estado de los procesos industriales
- Controlar equipos remotamente
- Monitorear alarmas y eventos
- Registrar datos históricos

### **¿Qué es ESP32?**

El ESP32 es un microcontrolador que incluye:

- Procesador de doble núcleo
- WiFi y Bluetooth integrados
- Múltiples pines de entrada/salida
- Capacidad de ejecutar MicroPython

## **Instalación Paso a Paso**

### **Paso 1: Instalar Git**

#### **En Windows:**

1. Ve a <https://git-scm.com/download/windows>
2. Descarga la versión más reciente (aproximadamente 50MB)
3. Ejecuta el instalador como administrador
4. En las opciones de instalación:

- Mantén las opciones por defecto
- Asegúrate de seleccionar "Git from the command line and also from 3rd-party software"

5. Completa la instalación

### Verificar instalación:

1. Presiona `Windows + R`
2. Escribe `cmd` y presiona Enter
3. En la ventana negra que aparece, escribe: `git --version`
4. Deberías ver algo como: `git version 2.41.0.windows.1`

### En Mac:

1. Presiona `Cmd + Espacio`
2. Escribe "Terminal" y presiona Enter
3. En la terminal, escribe: `git --version`
4. Si no está instalado, aparecerá una ventana pidiendo instalar las herramientas de desarrollo
5. Haz clic en "Instalar" y espera a que termine

### En Linux (Ubuntu/Debian):

```
bash
```

```
# Actualizar la lista de paquetes
```

```
sudo apt update
```

```
# Instalar Git
```

```
sudo apt install git
```

```
# Verificar instalación
```

```
git --version
```

## Paso 2: Instalar Python

### En Windows:

1. Ve a <https://www.python.org/downloads/>
2. Descarga Python 3.11 o superior
3. **IMPORTANTE:** Durante la instalación, marca la casilla "Add Python to PATH"

4. Selecciona "Install Now"

### Verificar instalación:

cmd

```
python --version  
pip --version
```

### En Mac:

bash

```
# Instalar usando Homebrew (recomendado)  
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"  
brew install python  
  
# Verificar  
python3 --version  
pip3 --version
```

### En Linux:

bash

```
sudo apt update  
sudo apt install python3 python3-pip  
python3 --version  
pip3 --version
```

## Paso 3: Instalar Thonny IDE (para ESP32)

1. Ve a <https://thonny.org>
2. Descarga la versión para tu sistema operativo
3. Instala siguiendo las instrucciones normales
4. Abre Thonny
5. Ve a "Tools" > "Options" > "Interpreter"
6. Selecciona "MicroPython (ESP32)"



## Descargando el Proyecto

## Método 1: Descarga Directa (Principiantes)

1. Ve a: <https://github.com/DanielTorresC14/Training-Didactic-PLC-WIFI>
2. Haz clic en el botón verde **"Code"**
3. Selecciona **"Download ZIP"**
4. Guarda el archivo en tu escritorio
5. Extrae el archivo ZIP (clic derecho > "Extraer aquí" o "Extract here")
6. Renombra la carpeta a `Training-Didactic-PLC-WIFI` (sin el "-main")

## Método 2: Usando Git (Recomendado)

### 1. Abrir terminal:

- Windows: Presiona `Windows + R`, escribe `cmd`, presiona Enter
- Mac: Presiona `Cmd + Espacio`, escribe "Terminal", presiona Enter
- Linux: Presiona `Ctrl + Alt + T`

### 2. Navegar a donde quieres guardar el proyecto:

```
bash
```

```
# Para ir al Escritorio
```

```
cd Desktop
```

```
# Para ir a Documentos
```

```
cd Documents
```

```
# Para crear una carpeta específica para proyectos
```

```
mkdir Proyectos
```

```
cd Proyectos
```

### 3. Clonar el repositorio:

```
bash
```

```
git clone https://github.com/DanielTorresC14/Training-Didactic-PLC-WIFI.git
```

### 4. Entrar al directorio del proyecto:

```
bash
```

```
cd Training-Didactic-PLC-WIFI
```

## 5. Ver el contenido:

```
bash
```

```
# En Windows
```

```
dir
```

```
# En Mac/Linux
```

```
ls -la
```

## Estructura del Proyecto

Basándome en la descripción del proyecto, la estructura típica sería:

```
Training-Didactic-PLC-WIFI/
├── README.md ..... # Información del proyecto
├── requirements.txt ..... # Dependencias de Python
├── ESP32/ ..... # Código para ESP32
│   ├── main.py ..... # Programa principal MicroPython
│   ├── boot.py ..... # Configuración de arranque
│   ├── wifi_config.py ..... # Configuración WiFi
│   └── plc_logic.py ..... # Lógica del PLC
├── HMI_Web/ ..... # Interfaz web
│   ├── app.py ..... # Aplicación web principal
│   ├── templates/ ..... # Plantillas HTML
│   │   ├── index.html
│   │   └── control.html
│   ├── static/ ..... # Archivos CSS, JS, imágenes
│   │   ├── css/
│   │   ├── js/
│   │   └── images/
│   └── config/
├── Documentation/ ..... # Documentación adicional
├── Examples/ ..... # Ejemplos de uso
└── LICENSE ..... # Licencia del proyecto
```

## Configuración del Entorno

### 1. Configurar el Entorno Python

bash

*# Navegar al directorio del proyecto*

`cd Training-Didactic-PLC-WIFI`

*# Crear un entorno virtual (recomendado)*

`python -m venv venv`

*# Activar el entorno virtual*

*# En Windows:*

`venv\Scripts\activate`

*# En Mac/Linux:*

`source venv/bin/activate`

*# Instalar dependencias*

`pip install -r requirements.txt`

## 2. Configurar ESP32

1. **Conectar el ESP32** a tu computadora via USB

2. **Abrir Thonny IDE**

3. **Configurar el intérprete:**

- Tools > Options > Interpreter
- Seleccionar "MicroPython (ESP32)"
- Seleccionar el puerto COM correcto

4. **Instalar MicroPython en ESP32** (si es necesario):

- En Thonny: Tools > Options > Interpreter
- Clic en "Install or update MicroPython"
- Seguir las instrucciones

## Configuración Inicial

### 1. Configurar WiFi en ESP32

Edita el archivo `ESP32/wifi_config.py`:

```
python
```

```
# Configuración WiFi
WIFI_SSID = "TU_NOMBRE_WIFI"
WIFI_PASSWORD = "TU_CONTRASEÑA_WIFI"
```

## 2. Configurar la Interfaz Web

Edita el archivo `HMI_Web/config/settings.py`:

```
python
```

```
# Configuración del servidor web
HOST = "0.0.0.0" # Permite acceso desde otras computadoras
PORT = 5000 # Puerto web
DEBUG = True # Modo desarrollo
```



## Ejecutando el Proyecto

### Paso 1: Programar el ESP32

1. **Abrir Thonny IDE**
2. **Conectar ESP32** via USB
3. **Abrir los archivos del ESP32:**
  - File > Open > Navegar a `ESP32/main.py`
4. **Cargar archivos al ESP32:**
  - Clic derecho en cada archivo Python
  - Seleccionar "Upload to /"
5. **Ejecutar el programa:**
  - Presionar el botón "Run" o F5

### Paso 2: Ejecutar la Interfaz Web



bash

*# Asegúrate de estar en el directorio del proyecto*

`cd Training-Didactic-PLC-WIFI`

*# Activar entorno virtual (si lo creaste)*

*# Windows:*

`venv\Scripts\activate`

*# Mac/Linux:*

`source venv/bin/activate`

*# Navegar al directorio HMI Web*

`cd HMI_Web`

*# Ejecutar la aplicación web*

`python app.py`

### Paso 3: Acceder a la Interfaz

1. **Abrir navegador web**
2. **Ir a:** <http://localhost:5000>
3. **Para acceder desde otros dispositivos:** [http://TU\\_IP:5000](http://TU_IP:5000)

### Usando la Interfaz HMI

#### Panel de Control Principal

- **Estados:** Visualiza el estado actual de entradas y salidas
- **Controles:** Botones para activar/desactivar salidas
- **Monitoreo:** Gráficos en tiempo real de sensores
- **Alarmas:** Notificaciones de eventos importantes

#### Funciones Típicas

- **Encender/Apagar LEDs**
- **Leer sensores de temperatura**
- **Controlar motores**
- **Monitorear entradas digitales**
- **Configurar temporizadores**

### Personalización

## Modificar la Lógica del PLC

Edita `ESP32/plc_logic.py`:

```
python

def control_logic():
    # Tu lógica personalizada aquí
    if input_1.value():
        output_1.on()
    else:
        output_1.off()
```

## Personalizar la Interfaz Web

1. **Modificar plantillas HTML:** `HMI_Web/templates/`
2. **Cambiar estilos CSS:** `HMI_Web/static/css/`
3. **Agregar funcionalidad JavaScript:** `HMI_Web/static/js/`

## Solución de Problemas Comunes

### El ESP32 no se conecta al WiFi

```
python

# Verificar configuración en wifi_config.py
# Asegurarse de que SSID y contraseña sean correctos
# Verificar que el router soporte 2.4GHz
```

### La interfaz web no carga

```
bash

# Verificar que Python esté corriendo
# Comprobar el puerto (generalmente 5000)
# Revisar firewall y antivirus
```

### Error al instalar dependencias

```
bash
```

```
# Actualizar pip  
python -m pip install --upgrade pip
```

```
# Instalar dependencias una por una  
pip install flask  
pip install socketio
```

## ESP32 no se detecta en Thonny

1. **Verificar drivers USB:** Instalar drivers CP2102 o CH340
2. **Comprobar cable USB:** Usar cable de datos, no solo de carga
3. **Verificar puerto COM:** Device Manager en Windows



## Recursos Adicionales

### Documentación MicroPython

- **Oficial:** <https://docs.micropython.org/>
- **ESP32 específico:** <https://docs.micropython.org/en/latest/esp32/>

### Tutoriales Recomendados

- **MicroPython con ESP32:** Tutoriales básicos de programación
- **Desarrollo de HMI:** Creación de interfaces industriales
- **Automatización con Python:** Conceptos avanzados de control

### Comunidades

- **Foro MicroPython:** <https://forum.micropython.org/>
- **Reddit r/esp32:** Comunidad activa de ESP32
- **GitHub Discussions:** En el repositorio del proyecto



## Contribuir al Proyecto

### Para Principiantes

1. **Reportar errores:** Crear "Issues" en GitHub
2. **Sugerir mejoras:** Usar "Discussions"
3. **Compartir experiencias:** Documentar tu uso del proyecto

## Para Desarrolladores

1. **Fork** el repositorio
2. **Crear rama** para tu función: `git checkout -b mi-nueva-funcion`
3. **Hacer commits**: `git commit -m "Agregar nueva función"`
4. **Push**: `git push origin mi-nueva-funcion`
5. **Crear Pull Request**

## Licencia y Créditos

- **Autor**: DanielTorresC14
- **Propósito**: Educativo y didáctico
- **Usar con**: Fines de aprendizaje y desarrollo

## Soporte

Si tienes problemas:

1. **Revisar esta documentación** completamente
2. **Buscar en Issues** del repositorio GitHub
3. **Crear nuevo Issue** si no encuentras solución
4. **Incluir detalles**: Sistema operativo, versión Python, mensajes de error exactos

## Próximos Pasos

Una vez que tengas el proyecto funcionando:

1. **Experimenta** con diferentes configuraciones
2. **Modifica** la lógica del PLC para tus necesidades
3. **Personaliza** la interfaz HMI
4. **Integra** sensores adicionales
5. **Documenta** tus modificaciones
6. **Comparte** tu experiencia con la comunidad

## Lista de Verificación Final

Antes de comenzar, asegúrate de tener:

- ☐ Git instalado y funcionando

- ☐ Python 3.7+ instalado
- ☐ Thonny IDE instalado
- ☐ ESP32 y cable USB disponibles
- ☐ Conexión WiFi configurada
- ☐ Proyecto descargado correctamente
- ☐ Entorno virtual creado (opcional pero recomendado)
- ☐ Dependencias instaladas

¡Ahora estás listo para comenzar tu aventura en el mundo de los PLCs didácticos con WiFi!

---

**Nota:** Esta documentación está diseñada para personas completamente nuevas en Git y programación. Si eres desarrollador experimentado, puedes saltar directamente a la sección de configuración técnica.