



**SISTEMAS EXPERTOS**

**PROFESOR: MAURICIO ALEJANDRO CABRERA  
ARELLANO**

**PROYECTO FINAL 3 PARCIAL - PLC**

**ALUMNO: PRECIADO MARTINEZ BRUNO AUGUSTO**

**MATRICULA :21110311**

## 1. Introducción

El presente documento detalla el funcionamiento de un sistema experto diseñado para recomendar un PLC (Controlador Lógico Programable) adecuado basado en las características proporcionadas por el usuario. Este sistema considera tres modelos específicos de PLC: Delta SA2, SIMATIC S7-1200 e INVT IVC1, y tiene en cuenta parámetros como el tipo de salidas, el protocolo de comunicación y la presencia de pantalla HMI.

## 2. Estructura del Proyecto

El proyecto se compone de dos partes principales:

1. Una base de datos MySQL donde se almacena la información de los PLCs.
2. Un script en Python que actúa como el motor de inferencia y proporciona la interfaz gráfica para la interacción del usuario.

## 3. Configuración de la Base de Datos

La base de datos, denominada `BASE1`, contiene una tabla llamada `plc` que almacena las especificaciones de cada modelo de PLC. La estructura de la tabla y los datos se insertan de la siguiente manera:

```
sql
Copiar código
CREATE DATABASE BASE1;

USE BASE1;

CREATE TABLE plc (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(255),
    salidas VARCHAR(50),
    protocolo_comunicacion VARCHAR(100),
    tipo_plc VARCHAR(50)
);

INSERT INTO plc (nombre, salidas, protocolo_comunicacion, tipo_plc)
VALUES
('Delta SA2', 'relevador', 'RS232', 'sin pantalla HMI'),
('SIMATIC S7-1200', 'transistor npn', 'S7 sobre ISO TCP RFC1006', 'sin
pantalla HMI'),
('INVT IVC1', 'transistor pnp', 'Mini DIN8 RS485', 'con pantalla HMI');
```

## 4. Descripción del Código

### 4.1. Conexión a la Base de Datos

La función `conectar_bd` se encarga de establecer una conexión con la base de datos MySQL. Utiliza las credenciales del usuario para acceder a la base de datos y maneja posibles errores de conexión:

```
python
Copiar código
import mysql.connector
from mysql.connector import errorcode

def conectar_bd():
    try:
        conexion = mysql.connector.connect(
            user='tu_usuario', # reemplazar con el usuario de MySQL
            password='tu_contraseña', # reemplazar con la contraseña de
MySQL            host='localhost',
            database='BASE1'
        )
        return conexion
    except mysql.connector.Error as err:
        if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
            print("Algo está mal con tu usuario o contraseña")
        elif err.errno == errorcode.ER_BAD_DB_ERROR:
            print("La base de datos no existe")
        else:
            print(err)
        return None
```

#### 4.2. Motor de Inferencia

La función `sistema_experto` realiza la consulta en la base de datos para encontrar un PLC que cumpla con las características especificadas por el usuario. Si se encuentra un PLC adecuado, se retorna su nombre; de lo contrario, se indica que no se encontró ningún PLC que cumpla con los requisitos.

```
python
Copiar código
def sistema_experto(salidas, protocolo_comunicacion, tipo_plc):
    conexion = conectar_bd()
    if conexion is None:
        return "No se pudo conectar a la base de datos."

    cursor = conexion.cursor()

    query = '''
        SELECT nombre FROM plc
        WHERE salidas = %s AND protocolo_comunicacion = %s AND tipo_plc =
%s
        LIMIT 1
    '''
    cursor.execute(query, (salidas, protocolo_comunicacion, tipo_plc))

    resultado = cursor.fetchone()
    conexion.close()
```

```

    if resultado:
        return f"Se recomienda utilizar el PLC: {resultado[0]}"
    else:
        return "No se encontró un PLC que cumpla con los requisitos especificados."

```

### 4.3. Interfaz Gráfica

Se utiliza la biblioteca `tkinter` para crear una interfaz gráfica de usuario (GUI) que permite seleccionar las características del PLC. Las opciones seleccionadas se capturan y se pasan a la función `sistema_experto` para obtener la recomendación.

```

python
Copiar código
import tkinter as tk
from tkinter import messagebox

def mostrar_recomendacion():
    salidas = salidas_var.get()
    protocolo_comunicacion = protocolo_comunicacion_var.get()
    tipo_plc = tipo_plc_var.get()

    recomendacion = sistema_experto(salidas, protocolo_comunicacion,
    tipo_plc)
    messagebox.showinfo("Recomendación del PLC", recomendacion)

# Crear la ventana principal
ventana = tk.Tk()
ventana.title("Sistema Experto para Elegir PLC")

# Variables para las opciones
salidas_var = tk.StringVar(value="relevador")
protocolo_comunicacion_var = tk.StringVar(value="RS232")
tipo_plc_var = tk.StringVar(value="sin pantalla HMI")

# Crear widgets
tk.Label(ventana, text="Seleccione las características del
PLC:").grid(row=0, columnspan=2, pady=10)

tk.Label(ventana, text="Salidas del PLC:").grid(row=1, column=0,
sticky='e')
tk.Radiobutton(ventana, text="Relevador", variable=salidas_var,
value="relevador").grid(row=1, column=1, sticky='w')
tk.Radiobutton(ventana, text="Transistor NPN", variable=salidas_var,
value="transistor npn").grid(row=2, column=1, sticky='w')
tk.Radiobutton(ventana, text="Transistor PNP", variable=salidas_var,
value="transistor pnp").grid(row=3, column=1, sticky='w')

tk.Label(ventana, text="Protocolo de Comunicación:").grid(row=4,
column=0, sticky='e')
tk.Radiobutton(ventana, text="RS232",
variable=protocolo_comunicacion_var, value="RS232").grid(row=4, column=1,
sticky='w')

```

```

tk.Radiobutton(ventana, text="S7 sobre ISO TCP RFC1006",
variable=protocolo_comunicacion_var, value="S7 sobre ISO TCP
RFC1006").grid(row=5, column=1, sticky='w')
tk.Radiobutton(ventana, text="Mini DIN8 RS485",
variable=protocolo_comunicacion_var, value="Mini DIN8 RS485").grid(row=6,
column=1, sticky='w')

tk.Label(ventana, text="Tipo de PLC:").grid(row=7, column=0, sticky='e')
tk.Radiobutton(ventana, text="Sin pantalla HMI", variable=tipo_plc_var,
value="sin pantalla HMI").grid(row=7, column=1, sticky='w')
tk.Radiobutton(ventana, text="Con pantalla HMI", variable=tipo_plc_var,
value="con pantalla HMI").grid(row=8, column=1, sticky='w')

tk.Button(ventana, text="Obtener Recomendación",
command=mostrar_recomendacion).grid(row=9, columnspan=2, pady=20)

# Ejecutar la aplicación
ventana.mainloop()

```

## 5. Reglas de la Lógica del Sistema Experto

El sistema experto sigue reglas específicas basadas en las características del PLC para proporcionar una recomendación adecuada:

1. **Delta SA2:**
  - Salidas: relevador
  - Protocolo de comunicación: RS232
  - Tipo de PLC: sin pantalla HMI
2. **SIMATIC S7-1200:**
  - Salidas: transistor npn
  - Protocolo de comunicación: S7 sobre ISO TCP RFC1006
  - Tipo de PLC: sin pantalla HMI
3. **INVT IVC1:**
  - Salidas: transistor pnp
  - Protocolo de comunicación: Mini DIN8 RS485
  - Tipo de PLC: con pantalla HMI

## 6. Conclusión

El sistema experto desarrollado proporciona una herramienta eficiente y fácil de usar para la selección de un PLC adecuado basado en las necesidades específicas del usuario. Utiliza una base de datos MySQL para almacenar y consultar la información de los PLCs y una interfaz gráfica de usuario para facilitar la interacción. Este enfoque permite una rápida y precisa recomendación de PLC, mejorando la toma de decisiones en entornos industriales y de automatización.

## ENLACE DE GITHUB DEL PROYECTO

<https://github.com/martinez2331/Sistemas-Expertos-Proyecto-final>