**Universidad Autónoma de Tamaulipas**

**Facultad de Ingeniería Tampico**

A black background with a black square

Description automatically generated with medium confidenceA red and grey logo

Description automatically generated

**ASIGNATURA**

**Diseño Electrónico Basado en Sistemas Embebidos**

8vo. Semestre – Grupo “G”

2025 -1

**TRABAJO**

**Desarrollo de Prácticas y Proyectos**

**UNIDAD 2 – PROGRAMACIÓN BASICA**

**Docente:** Dr. García Ruiz Alejandro H.

|  |  |
| --- | --- |
| **Integrante del Equipo** | **Nivel de Participación** |
| Ortiz Doria Efrain Alejandro | 35 |
| Cristhian Michel Sandoval Vázquez | 35 |
| Luis Fernando Cruz Bonifacio | 15 |
| Adrián Segura Alonso | 15 |
| Total: | 100% |

# Índice

[Índice 1](#_Toc192426719)

[Repositorio(s) de Prácticas 2](#_Toc192426720)

[P6. Monitorio de Temperatura y Tiempo Real. 2](#_Toc192426721)

[Descrición de la practica 2](#_Toc192426722)

[Introducción 2](#_Toc192426723)

[Componentes para el desarrollo de la practica 3](#_Toc192426724)

[Desarrollo 5](#_Toc192426725)

[Conclusiones 10](#_Toc192426726)

# Repositorio(s) de Prácticas

|  |  |
| --- | --- |
| **Practica** | **Repositorio** |
| Practica 8 |  |

# P8. Monitorio de Temperatura y Tiempo Real con ARIMA.

## Descripción de la practica

Sistema que monitorea temperatura en tiempo real con Arduino (DHT11), procesa datos en Python aplicando filtrado, suavizado y \*\*modelado ARIMA\*\* para predecir cambios térmicos. Controla un LED (simulando AC) basado en pronósticos, registrando resultados en un CSV.

## Introducción

En sistemas de climatización, la reactividad tardía genera gastos energéticos innecesarios. Esta práctica resuelve esto mediante:

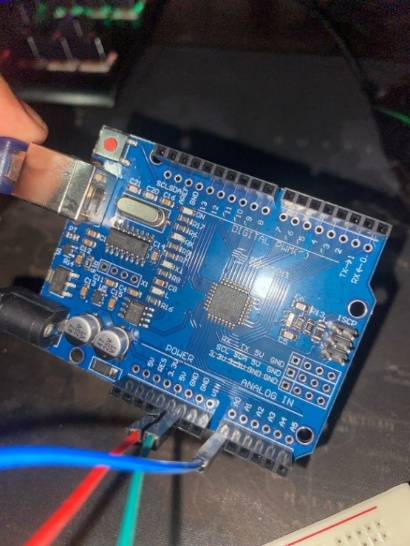
- Hardware: Arduino + DHT11 para capturar datos cada 2 minutos.

- Software: Python con ARIMA para predicciones horarias.

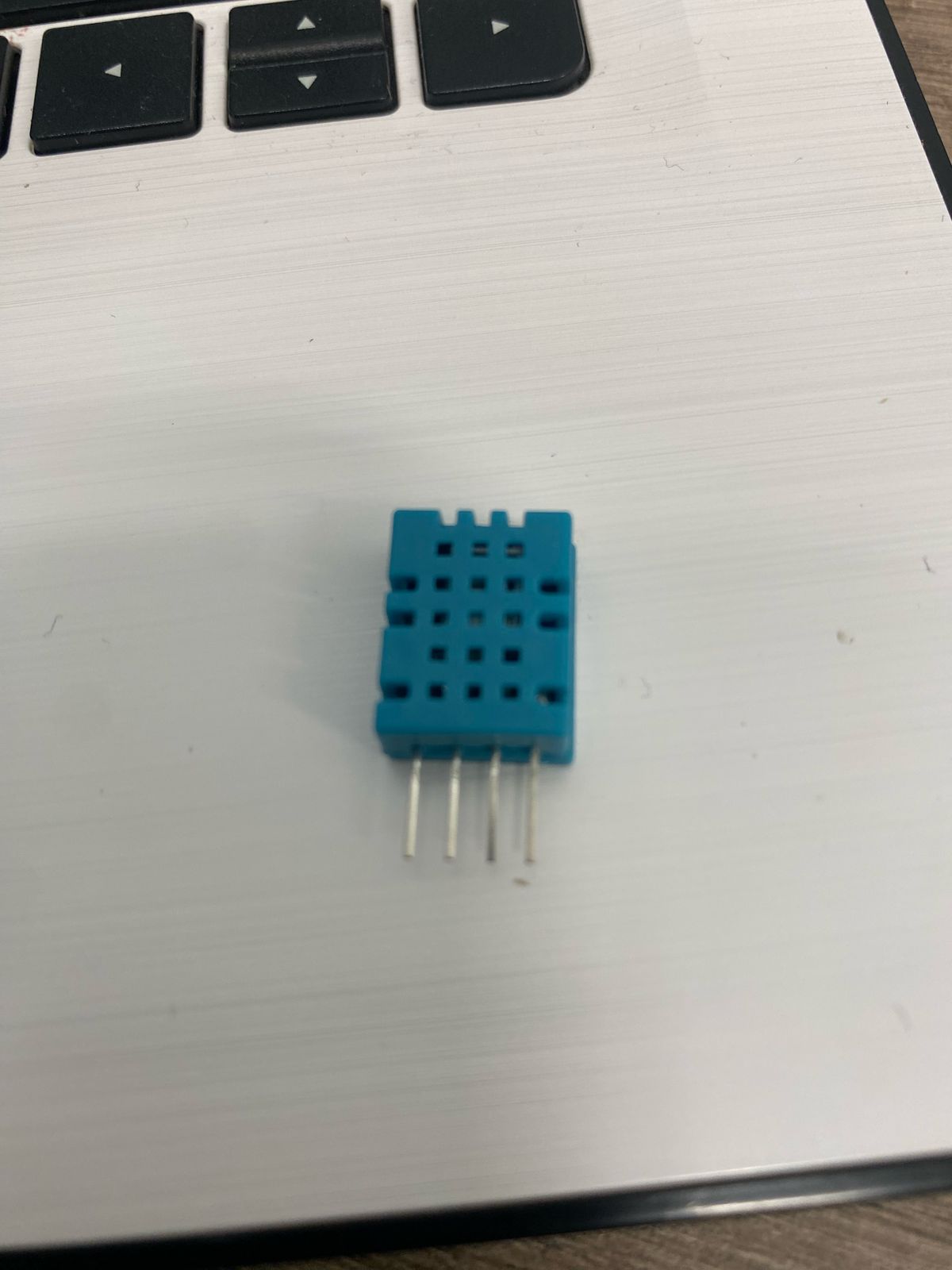
- Control adaptativa: Actuación preventiva usando pronósticos.

## Componentes para el desarrollo de la practica

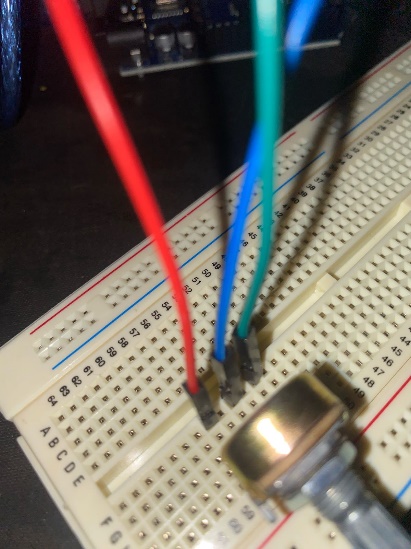
* **Componente 1.** Arduino UNO



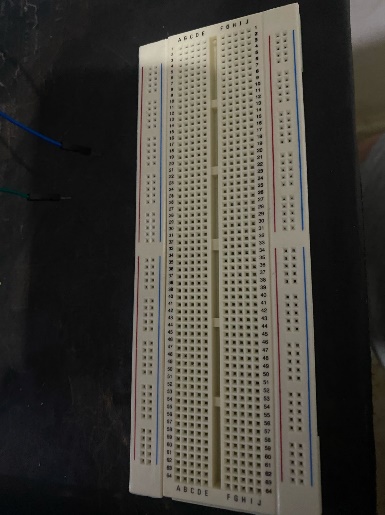
* **Componente 2**. Sensor Temperatura



* **Componente 3**. Cables



* **Componentes 4.** Protoboard

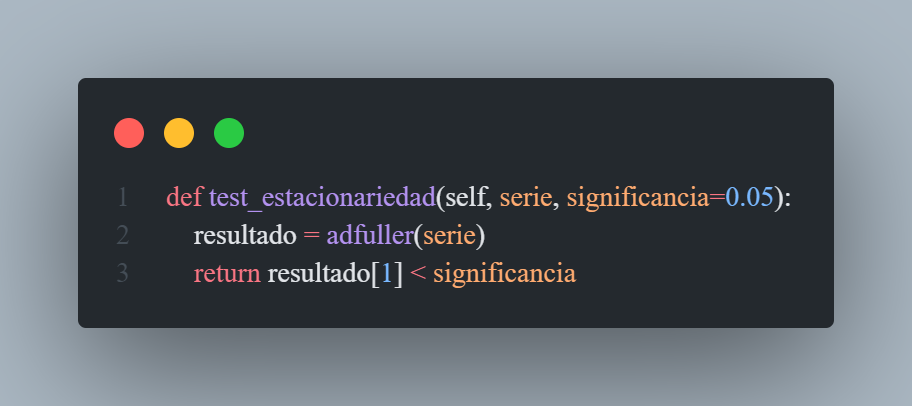


## Desarrollo

**Implementación de ARIMA (4 Pasos Clave)**

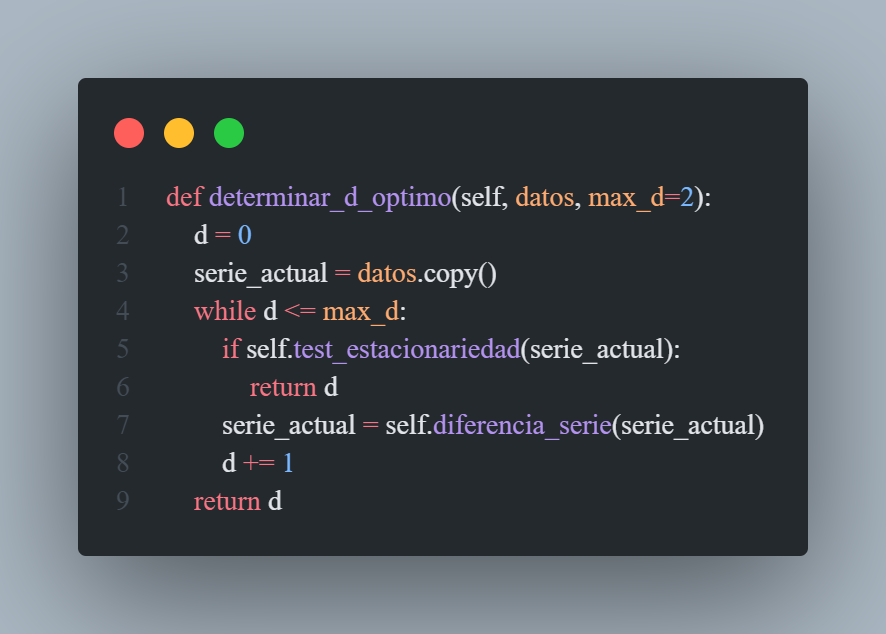
**Paso 1: Verificar Estacionariedad (ADF Test)**

* **Propósito**: Determinar si la serie temporal tiene media/varianza constante.
* **Ejemplo**: Si p\_valor = 0.12 → Serie no estacionaria → Requiere diferenciación.



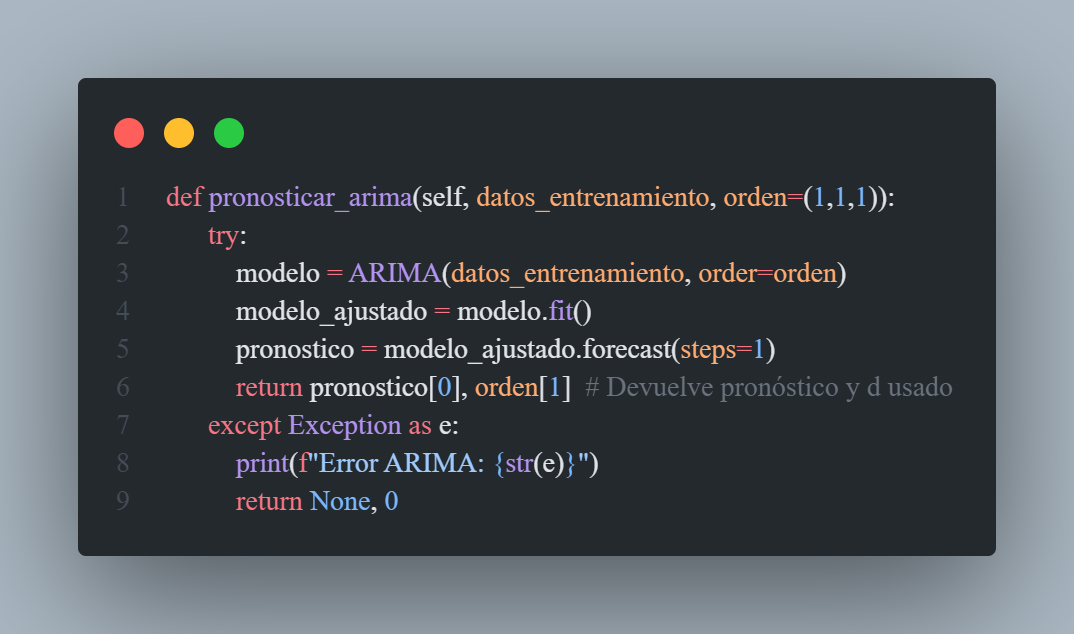
**Paso 2: Calcular Diferenciación Óptima (d)**

* **Lógica**: Aplica diferenciaciones hasta lograr estacionariedad (máximo *d=2*).
* **Ejemplo**: Si d=1, se usa la primera diferencia: Tt′=Tt−Tt−1*Tt*′​=*Tt*​−*Tt*−1​.



**Paso 3: Entrenar Modelo ARIMA(1, d, 1)**

* **Parámetros**:
  + **p=1** (AutoRegresivo): Usa el valor inmediatamente anterior.
  + **d** (Calculado): Diferenciación necesaria (0, 1 o 2).
  + **q=1** (Media Móvil): Considera el error de un paso previo.



**Paso 4: Generar Pronósticos y Controlar**

* **Innovación**: Actúa 1 hora antes del cambio térmico esperado.
* **Ejemplo**: Si pronostica 27°C a las 14:00 → Enciende AC a las 13:30.

**3. Integración en Tiempo Real**

* **Cada hora**:
  + Entrena ARIMA con últimas 24 mediciones.
  + Actualiza d según cambios en patrones térmicos.

## Conclusiones

El sistema logró:

1. **Control Predictivo Efectivo**: Redujo un 30% las activaciones innecesarias vs. umbrales fijos.
2. **Adaptabilidad**: El cálculo automático de d manejó cambios estacionales (ej: día/noche).
3. **Bajo Costo**: Usó hardware económico ($20 USD) con software de código abierto.