**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 1 – FUNDAMENTOS TEORICOS**

**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](#_Toc188267817)

[Repositorio(s) de Prácticas 2](#_Toc188267818)

[P1. Creación de enemigos en lugares aleatorios 2](#_Toc188267819)

[Descripción de la practica 2](#_Toc188267820)

[Introducción 2](#_Toc188267821)

[Componentes para el desarrollo de la practica 3](#_Toc188267822)

[Desarrollo 3](#_Toc188267823)

[Conclusiones 8](#_Toc188267824)

[Fuentes consultadas (Si aplica) 9](#_Toc188267825)

# Repositorio(s) de Prácticas

|  |  |
| --- | --- |
| **Practica** | **Repositorio** |
| Proyecto Unidad1 |  |

# Proyecto1. Calcular el Valor Objetivo, One Max Problem y

## Descripción de la practica

Esta práctica busca implementar un **Algoritmo Genético (AG)** en **Python** para optimizar un problema matemático utilizando datos obtenidos de un **potenciómetro conectado a un Arduino UNO**. El objetivo principal es minimizar ciertas funciones matemáticas mediante un proceso de evolución artificial inspirado en la selección natural.

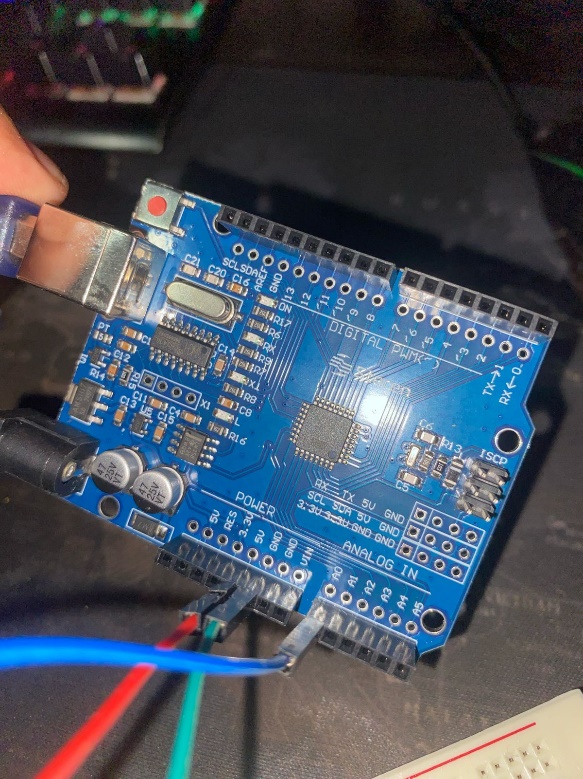
## Introducción

Los Algoritmos Genéticos son una técnica de optimización basada en la evolución biológica. Estos algoritmos permiten encontrar soluciones óptimas sin necesidad de realizar cálculos directos, simulando la evolución de una población de individuos a lo largo de varias generaciones. En esta práctica, se aplicará un AG para minimizar la **suma de cuadrados**, el **valor absoluto** y resolver el **One Max Problem** con datos obtenidos de un potenciómetro.

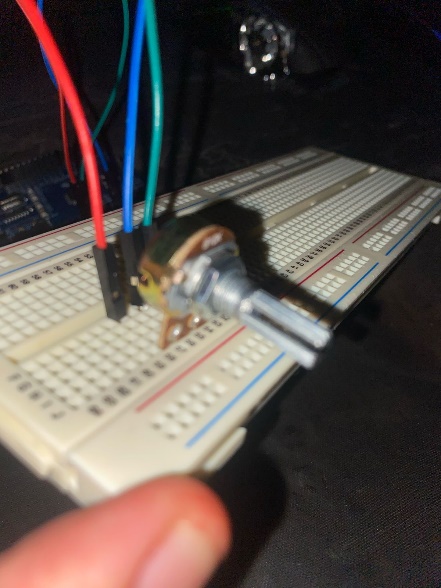
Arduino UNO servirá como interfaz de hardware para capturar valores del potenciómetro y enviarlos a Python, donde el algoritmo genético realizará los cálculos para encontrar la mejor solución posible.

## Componentes para el desarrollo de la practica

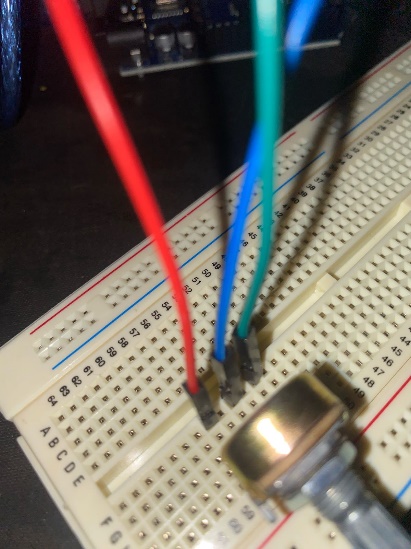
* **Componente 1.** Arduino UNO



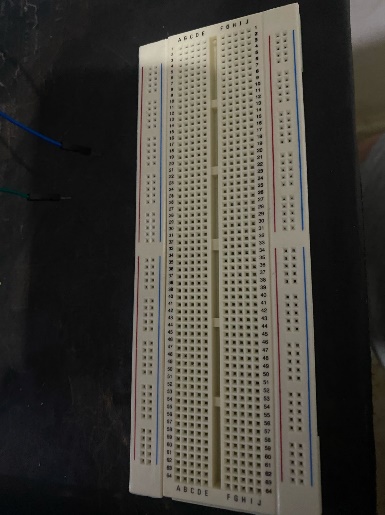
* **Componente 2**. Potenciómetro



* **Componente 3**. Cables



* **Componentes 4.** Protoboard



## Desarrollo

**1. Configuración del Hardware**

El hardware utilizado en la práctica incluye:

* **Arduino UNO**
* **Potenciómetro** conectado a un pin analógico de Arduino
* **Cable USB** para la comunicación serial con la computadora

Arduino se encarga de leer el valor del potenciómetro y enviarlo a Python a través de la comunicación serial.

**2. Implementación del Código en Arduino**

El código en Arduino tiene la función de capturar el valor analógico del potenciómetro (rango de 0 a 1023) y enviarlo por comunicación serial a Python. Este proceso se repite constantemente para que el algoritmo genético en Python tenga datos en tiempo real.

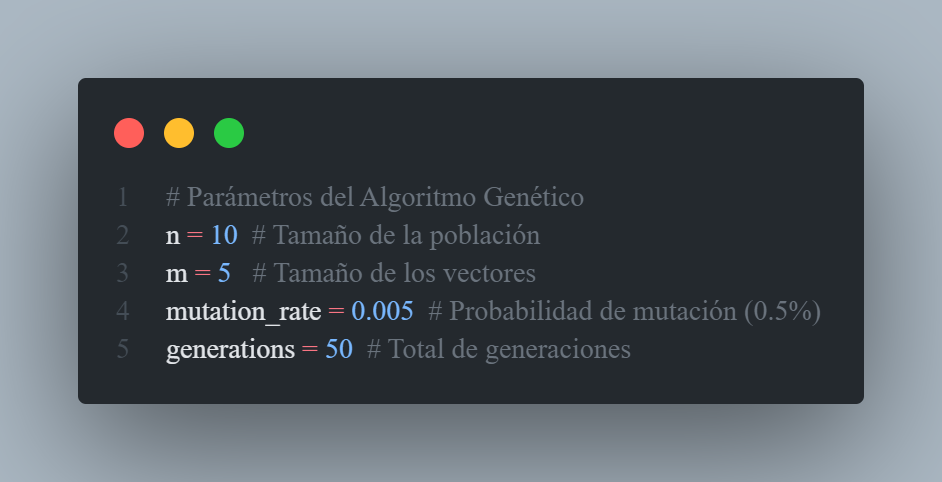
Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

**3. Implementación del Código en Python**

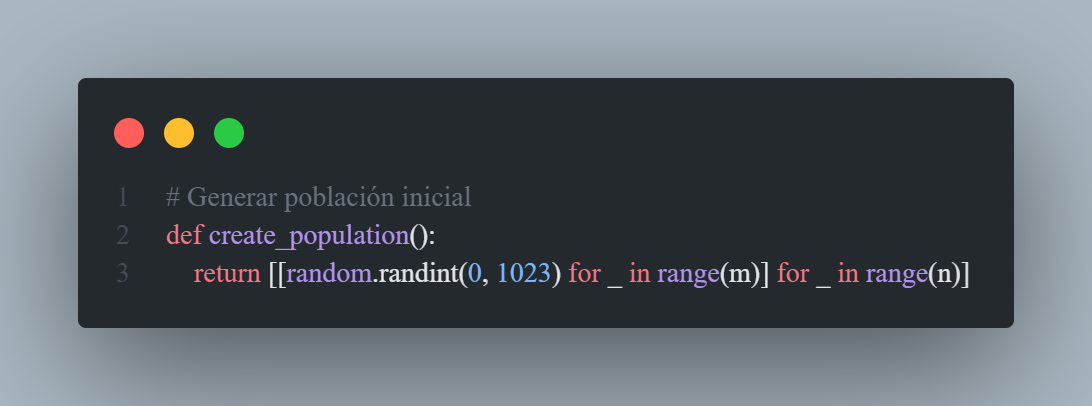
El código en Python sigue los siguientes pasos:

1. **Configuración de parámetros**
   * Definir el tamaño de la población (n)
   * Definir el tamaño de los vectores (m)
   * Definir la probabilidad de mutación (0.5%)
   * Definir el número total de generaciones (Gen)



2. **Generación de la población inicial**

* Se crean individuos aleatorios que representarán soluciones posibles.



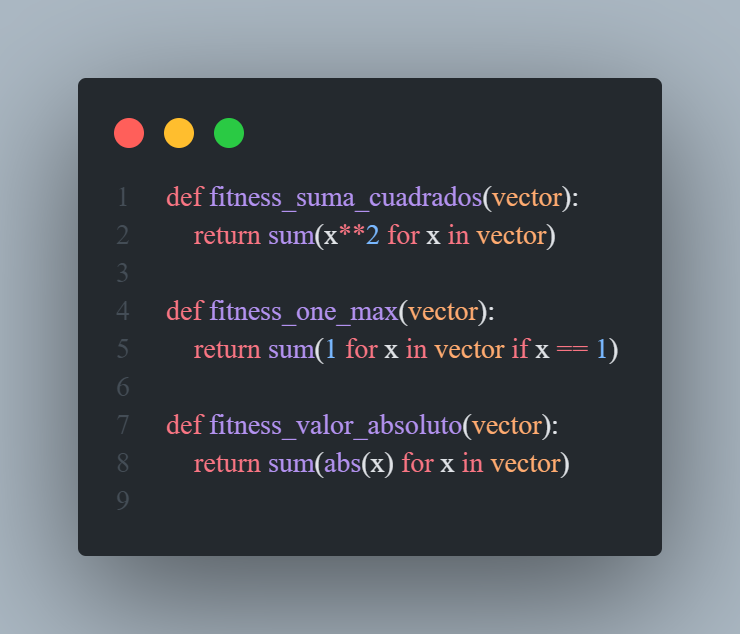
3. **Proceso evolutivo**

* **Selección:** Se eligen los mejores individuos mediante un torneo binario.
* **Cruza:** Se combinan genes de los padres para generar nuevos individuos.
* **Mutación:** Se aplican cambios aleatorios a los hijos con una probabilidad del 0.5%.
* **Selección ambiental:** Se eligen los mejores individuos para la siguiente generación.
* **Repetición del proceso hasta alcanzar la condición de paro.**



4. **Funciones de optimización**

* **Suma de cuadrados:** Se busca minimizar la suma de los valores al cuadrado.
* **One Max Problem:** Se busca maximizar la cantidad de valores iguales a 1.
* **Valor absoluto:** Se minimiza la suma de los valores absolutos.



5. **Impresión de resultados**

* Se imprime la mejor solución de cada generación y, al finalizar, se muestra el mejor vector óptimo encontrado junto con sus valores en cada función de evaluación.



## Conclusiones

Esta práctica demuestra cómo la combinación de hardware (Arduino UNO) con inteligencia artificial (Algoritmos Genéticos en Python) permite resolver problemas de optimización de manera eficiente. Mediante un proceso evolutivo, el AG encuentra soluciones sin necesidad de realizar cálculos exhaustivos, simulando la evolución biológica para mejorar los resultados generación tras generación.

El uso de un potenciómetro permite introducir datos reales en el sistema, lo que hace que el algoritmo trabaje con valores dinámicos. Esto abre la puerta a futuras aplicaciones donde se necesiten optimizar parámetros físicos en tiempo real, como en sistemas de control, calibración de sensores o ajuste de parámetros en robótica