7. Programe el anterior problema en Python con

| **CÓDIGO DE COLAB** |
| --- |
| #ENLAZAMOS DRIVE  from google.colab import drive  drive.mount("/content/Drive")  #IMPORTAMOS DATOS  import pandas as pd  data = pd.read\_csv("/content/Drive/MyDrive/datos/ejercicio7.csv")  print(data) |
| **CORRIDA EN PANTALLA** |
|  |

a. El uso de DEAP

| **CÓDIGO DE COLAB** |
| --- |
| #USAMOS DEAP  !pip install deap |
| #RESOLVEMOS EJERCICIO CON DEAP  import array  import random  import math  import csv # Para leer el archivo CSV  import numpy  from deap import algorithms  from deap import base  from deap import creator  from deap import tools  # Definimos la clase Fitness y el tipo de Individuo  creator.create("FitnessMax", base.Fitness, weights=(1.0,))  creator.create("Individual", array.array, typecode='b', fitness=creator.FitnessMax)  toolbox = base.Toolbox()  # Generador de atributos (binario)  toolbox.register("attr\_bool", random.randint, 0, 1)  # Inicialización de estructura  toolbox.register("individual", tools.initRepeat, creator.Individual, toolbox.attr\_bool, 4) # 4 bits por individuo  toolbox.register("population", tools.initRepeat, list, toolbox.individual)  # Cargar los datos del CSV  def cargar\_datos\_csv(filename):  datos = []  with open(filename, mode='r') as file:  reader = csv.DictReader(file)  for row in reader:  datos.append(row)  return datos  # Función de evaluación con la fórmula f(x) = x^2 \* x - 1  def fevaluacion2(individual):  # Convertir el individuo de binario a decimal  elevado = 0  suma = 0  for i in reversed(individual):  suma = suma + i \* math.pow(2, elevado)  elevado = elevado + 1  # Aplicar la fórmula f(x) = x^2 \* x - 1  resultado = (suma \*\* 2) \* suma - 1  return resultado,  # Registrar la función de evaluación, cruce, mutación y selección  toolbox.register("evaluate", fevaluacion2)  toolbox.register("mate", tools.cxTwoPoint)  toolbox.register("mutate", tools.mutFlipBit, indpb=0.05)  toolbox.register("select", tools.selTournament, tournsize=3)  def main():  # Cargar los datos desde el CSV  datos = cargar\_datos\_csv("/content/Drive/MyDrive/datos/ejercicio7.csv")  # Inicializar la población  pop = toolbox.population(n=len(datos)) # Número de individuos igual al número de filas en el CSV  hof = tools.HallOfFame(1)  stats = tools.Statistics(lambda ind: ind.fitness.values)  stats.register("avg", numpy.mean)  stats.register("std", numpy.std)  stats.register("min", numpy.min)  stats.register("max", numpy.max)  # Ejecutar el algoritmo evolutivo  pop, log = algorithms.eaSimple(pop, toolbox, cxpb=0.5, mutpb=0.2, ngen=3,  stats=stats, halloffame=hof, verbose=False)  # Imprimir encabezado  print(f"{'Original':<10}{'Población':<12}{'f(x)':<15}{'Fenotipo':<10}{'pdiv14':<10}{'pdiv24':<10}{'Cruce':<10}{'Población Final':<15}")  print("-" \* 92)  # Mostrar los resultados de la población final en columnas alineadas  for ind, row in zip(pop, datos):  # Convertimos el individuo en su valor decimal  fenotipo = int("".join(map(str, ind)), 2)  f\_x = fevaluacion2(ind)[0] # Evaluamos f(x) = x^2 \* x - 1  print(f"{row['Original']:<10}{row['poblacion']:<12}{f\_x:<15}{fenotipo:<10}{row['pdiv14']:<10}{row['pdiv24']:<10}{row['Cruce']:<10}{row['Poblacionfinal']:<15}")  return pop, log, hof  if \_\_name\_\_ == "\_\_main\_\_":  pop, log, hof = main()  print("---")  print(hof)  print("---")  print(log) |
| **CORRIDA DE PANTALLA** |
|  |

b. Sin el uso de DEAP

| **CÓDIGO EN COLAB** |
| --- |
| import random  import math  import csv # Para leer el archivo CSV  # Cargar los datos del CSV  def cargar\_datos\_csv(filename):  datos = []  with open(filename, mode='r') as file:  reader = csv.DictReader(file)  for row in reader:  datos.append(row)  return datos  # Función de evaluación con la fórmula f(x) = x^2 \* x - 1  def fevaluacion2(individual):  # Convertir el individuo de binario a decimal  elevado = 0  suma = 0  for i in reversed(individual):  suma = suma + i \* math.pow(2, elevado)  elevado = elevado + 1  # Aplicar la fórmula f(x) = x^2 \* x - 1  resultado = (suma \*\* 2) \* suma - 1  return resultado  def main():  # Cargar los datos desde el CSV  datos = cargar\_datos\_csv("/content/Drive/MyDrive/datos/ejercicio7.csv")  # Inicializar la población  pop = []  for \_ in range(len(datos)): # Número de individuos igual al número de filas en el CSV  individuo = [random.randint(0, 1) for \_ in range(4)] # 4 bits por individuo  pop.append(individuo)  # Imprimir encabezado  print(f"{'Original':<10}{'Población':<12}{'f(x)':<15}{'Fenotipo':<10}{'pdiv14':<10}{'pdiv24':<10}{'Cruce':<10}{'Población Final':<15}")  print("-" \* 92)  # Mostrar los resultados de la población final en columnas alineadas  for ind, row in zip(pop, datos):  # Convertimos el individuo en su valor decimal (fenotipo)  fenotipo = int("".join(map(str, ind)), 2)  f\_x = fevaluacion2(ind) # Evaluamos f(x) = x^2 \* x - 1  print(f"{row['Original']:<10}{row['poblacion']:<12}{f\_x:<15}{fenotipo:<10}{row['pdiv14']:<10}{row['pdiv24']:<10}{row['Cruce']:<10}{row['Poblacionfinal']:<15}")  if \_\_name\_\_ == "\_\_main\_\_":  main() |
| **CORRIDA DE PANTALLA** |
|  |