

```

#Inteligência Computacional
#Algoritmo Evolutivo

import time
import random

class evolucao:
    def __init__(self, objetivo):
        self.objetivo = objetivo.upper()
        self.genes = "ABCDEFGHJKLMNOPQRSTUVWXYZ "
        self.populacao = self.gerar_individuo()
        self.geracao = 0

    def gerar_individuo(self):
        return "".join(random.choice(self.genes) for _ in
range(len(self.objetivo)))

    def calcular_pontos(self, dna):
        pontos = sum(1 for i, letra in enumerate(dna) if letra ==
self.objetivo[i])
        return (pontos / len(self.objetivo)) * 100

    def mutacao(self, dna):
        lista = list(dna)
        pos = random.randint(0, len(self.objetivo) - 1)
        lista[pos] = random.choice(self.genes)
        return "".join(lista)

    def executar(self, delay = 0.1):
        print(f"Iniciar evolução até {self.objetivo}\n")

        while self.populacao != self.objetivo:
            self.geracao += 1
            descendente = self.mutacao(self.populacao)

            if self.calcular_pontos(descendente) >=
self.calcular_pontos(self.populacao):
                self.populacao = descendente

            acuracia = self.calcular_pontos(self.populacao)
            print(f"Geração: {self.geracao} | DNA: {self.populacao} |
Acurácia: {acuracia}")
            time.sleep(delay)

        print(f"\n Evolução concluída em {self.geracao} gerações!")

if __name__ == "__main__":
    experimento = evolucao("MATHAUS")
    experimento.executar()

```

