



A Doninha de Dawkins

A teoria da evolução exemplificada em um programa

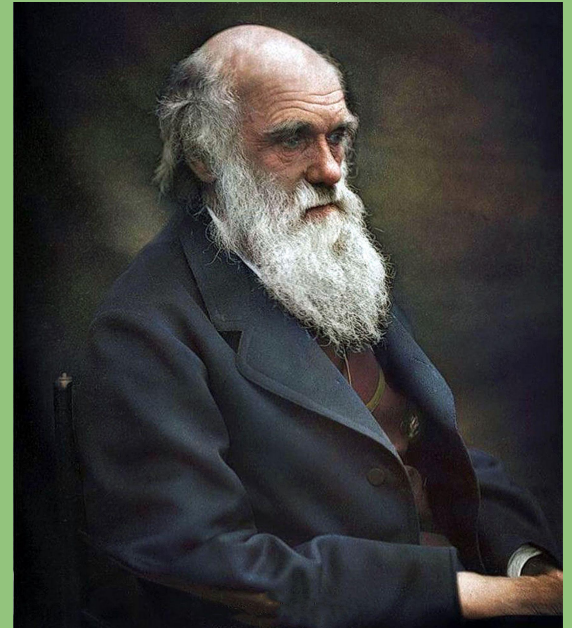
Equipe:

- Gabriel Alexander Farias de Lima Teixeira
- Helder Melik Schramm
- Natan Siqueira dos Santos
- Vinícius Soares da Costa
- Victor Yan Pereira e Lima



Primeiramente, o que é a Teoria da Evolução?

- Desenvolvida por Charles Darwin durante muitos anos de estudo prático e muitas correlações entre fósseis de animais extintos e animais modernos, ela explica como as espécies mudam ao longo do tempo: a sobrevivência na natureza se dá pelas espécies mais adaptadas, e não pelas mais fortes. É a evolução das espécies.





E qual a relação com o programa de Dawkins?

- De acordo com Richard Dawkins, sistemas evolucionários acontecem baseados em com variações aleatórias combinadas com seleções cumulativas não aleatórias - e que não acontecem por pura chance.
- O programa tem como objetivo simular esse processo: suponha que um macaco tenha que escrever a frase “Methinks it is like a weasel”, da obra Hamlet de Shakespeare, de forma randômica em um teclado com 26 letras e Espaço.
- O programa tentará recriar a frase fazendo cópias da sua primeira iteração, porém com erros aleatórios, de forma que com o passar das gerações, a frase original estará feita.

Methinks it is like a weasel.

Com a frase original formada, o programa consegue provar que a evolução não se dá por chances e combinações aleatórias sem sentido, mas com mutações que possuem diferenças cumulativas.

Portanto, na natureza essas mudanças podem ser vistas pela adaptação aos ambientes e peculiaridades nas espécies, como resistência a climas áridos, pescoços longos, melhor visão, entre outros.





E na programação, como fica?



Java

```
// Método principal do programa
public static void runWeasel() {
    System.out.println("Starting...\n");

    WParent theParent = new WParent( TARGET.length() );
    WParent[] offspring = new WParent[NEW_WEASELS];
    int generations = 1;
    int bestAccuracy = theParent.accuracy();

    System.out.println("geração: maior acurácia: genoma mais acurado");
    System.out.println(generations + " : " + bestAccuracy + " : " + theParent.getGenome() );

    // Cria uma nova geração até alcançar a String Alvo
    while( bestAccuracy < TARGET.length() ) {
        generations++;

        // Cria um novos Pais/Filhos para essa geração
        for(int i=0; i<NEW_WEASELS; i++) {
            offspring[i] = new WParent( theParent.getGenome() );
        }

        // Verifica qual o melhor resultado da geração para gerar filhos
        theParent = offspring[0]; // Index Pai/Filho de partido
        bestAccuracy = offspring[0].accuracy(); // Index de Acurácia de Partida
        for(int i=1; i<NEW_WEASELS; i++) {
            if( offspring[i].accuracy() > bestAccuracy ) {
                theParent = offspring[i];
                bestAccuracy = offspring[i].accuracy();
            }
        }

        // Imprimindo resultado da geração
        System.out.println(generations + " : " + bestAccuracy + " : " + theParent.getGenome() );
    }
}
```



C

```
int cont = 1;
while (strcmp(target, random))
{
    char copies[100][29];
    int bigger_value = 0;
    int bigger_value_idx = 0;

    for (int i = 0; i < 100; i++)
    {
        int value = 0;
        strcpy(copies[i], random);
        for (int j = 0; j < 29; j++)
        {
            if (rand() % 100 < 5)
            {
                copies[i][j] = alphabet[rand() % 27];
            }
            if (copies[i][j] == target[j])
            {
                value++;
            }
        }

        if (value > bigger_value)
        {
            bigger_value = value;
            bigger_value_idx = i;
        }
    }
    strcpy(random, copies[bigger_value_idx]);
    printf("Actual String: %s, Generation: %d\n", random, cont);
    cont++;
}
```




Python

```
alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ "
key_string = "METHINKS IT IS LIKE A WEASEL"
random_str = ""

for i in range(0, 28):
    random_str = random_str + alphabet[randint(0, 26)]

cont = 1
while random_str != key_string:
    copies = [random_str] * 100
    bigger_value = 0
    bigger_value_string = random_str

    for i in range(0, 100):
        value = 0
        string_list = []
        for j in range(0, 28):
            string_list.append(copies[i][j])
        for j in range(0, 28):
            if(randint(1, 100) <= 5):
                string_list[j] = alphabet[randint(0, 26)]
            if(string_list[j] == key_string[j]):
                value += 1
        copies[i] = "".join(string_list)
        if(value > bigger_value):
            bigger_value = value
            bigger_value_string = copies[i]

    random_str = bigger_value_string

    system("clear")
    print("Target string:", key_string)
    print("Actual String: {} \nGeneration: {}".format(random_str, cont))
    cont += 1
```



E na programação, como fica?

| | Python | Java | C |
|---------------|--------|-------|-------|
| Implementação | Boa | Média | Ruim |
| Legibilidade | Média | Boa | Média |
| Tamanho | Baixo | Alto | Médio |



Obrigado pela atenção!

