Introdução aos Algoritmos com Rust

https://github.com/ricardodarocha/Rust/blob/main/Curso/Rust.md

Estrutura Básica de um Programa em Rust

Um programa em Rust sempre começa pela função main:

```rust
fn main() {
    println!("Olá, mundo!");
}
```

▶️ executar

Exercício

```
Imprima! "Olá {fulano}" | fulano = seu_nome
```

Variáveis

```
    let nome = "Aluno";
    println!("Olá, {aluno}", aluno = nome);
```

▶️ executar

Tipos Primitivos

Rust possui diversos tipos de dados primitivos:

$\mathbb{N}$ Números naturais (u32, u64)

$\mathbb{Z}$ Números Inteiros (i32, i64)

$\mathbb{R}$ Real, Double (f32, f64)

Booleano (bool)

Caracteres (char)

```
let inteiro: i32 = 10;
let flutuante: f32 = 3.14;
let mol = 6.02e23;
let booleano: bool = true;
let caractere: char = 'A';
```

▶️ executar

Declaração de variáveis

```
let nome = "Santos Dummont";
let nascimento = 1873;
let idade = 2025 - nascimento;
```

▶️ executar

Exercício 2

Faça o programa imprimir 14 bis. Percorra um fluxo de repetição de 1 a 14,
incrementando um contador e imprima o número atual,
até que na última iteração o programa imprima 14 bis

```
for i in 1..=14 {
    imprima!("{} bis", i);
}
```

Variáveis mutáveis

```
let contador = 1; ❌
let mut contador = 1; ✅
contador += 1;
```

Tipo String

```
let mut jogador: String;

jogador = "Elfo".to_string();
jogador = "Bruxo".to_string();
```

▶️ executar

Blocos de decisão

```
if x == 42 {

}
```

Exemplo

```rust
fn main() {
    let idade = 26;
    if idade > 18 {
        println!("Maior de idade");
    } else
        println!("Menor de idade");
    }
```

Fluxo de repetição

```
fn main() {
    let mut contador = 0;
    loop {
        if contador >= 5 {
            break;
        }
        println!("Contador: {}", contador);
        contador += 1;
    }
}
```

For

```rust
fn main() {
    for i in 1..=5 {
        println!("Número: {}", i);
    }
}
```

Matrizes Array

```
let alfabeto: [char; 3] = ['a','b','c'];
```

▶️ executar