



# Variáveis



# Variáveis

Rust, uma linguagem de programação focada em segurança e desempenho, possui um sistema de gerenciamento de memória inovador e robusto. O entendimento das variáveis e da mutabilidade é fundamental para trabalhar eficientemente com Rust.

Em Rust, as variáveis são declaradas utilizando a palavra-chave **let**. O tipo da variável pode ser explicitamente indicado ou inferido pelo compilador.



# Mutabilidade

A mutabilidade em Rust é explicitamente controlada. Por padrão, as variáveis são imutáveis, o que significa que não podem ser modificadas após a sua primeira atribuição.

Para tornar uma variável mutável, utiliza-se a palavra-chave **mut**. O controle rigoroso sobre a mutabilidade em Rust contribui para a segurança do código. A prevenção contra mutações não autorizadas ajuda a evitar erros comuns, como race conditions e data races.



# Propriedade Única (Ownership)

Rust introduz o conceito de propriedade única para gerenciar a alocação de memória. Cada valor tem exatamente um "proprietário". Quando proprietário sai de escopo, o valor é liberado automaticamente.

```
let texto = String::from("hello");  
  
let texto2 = texto;
```



# Referências e Empréstimos

Para referenciar valores sem transferir a propriedade, Rust utiliza referências. Elas podem ser mutáveis ou imutáveis.

```
let mut valor = 50;  
let referencia = &valor;  
  
let referencia_mut = &mut valor;
```



# Conclusão

O sistema de variáveis e mutabilidade em Rust proporciona um equilíbrio entre desempenho e segurança. A ênfase na prevenção de erros comuns, combinada com a flexibilidade controlada, torna Rust uma escolha poderosa para desenvolvedores que buscam eficiência e confiabilidade em seus programas.



# Saiba mais, acesse:



<https://hcode.com.br>

