
**UNIVERSIDADE FEDERAL DE SERGIPE**

**Campus Professor Alberto Carvalho**

**Departamento de Sistemas de Informação – DSI**

**Discentes: Bruna Keila Oliveira Santos**

**Eduardo Santos Santana Bispo**

**Janaina Ferreira Santos**

**Thalyta Costa dos Santos**

**Docente: André Luís Menezes**

**Disciplina: Linguagens formais e tradutores**

# Linguagem de programação: Go

## 1. Introdução

A linguagem de programação Go foi criada pela Google em 2007, com a ajuda de Robert Griesemer, Rob Pike e Ken Thompson, com o propósito de ser uma linguagem mais simples, com bom desempenho, escalabilidade e de fácil manutenção, em 2009 foi lançada em código aberto.

Esta linguagem é compilada, o seu código-fonte é executado diretamente pelo sistema operacional, diferentemente de linguagens interpretadas, como Java. Além disso, ela é estaticamente tipada, ou seja, suas variáveis tem seus tipos definidos no código do programa. Ela também permite programação simultânea e execução adiada de algumas funções e possui Garbage Collector Nativo.

## 2. Lexemas do Go

Utilizamos, em Go, as seguintes regras para criação de um identificador:

- Os nome de identificadores devem consistir em apenas uma palavra, sem espaços em branco
- Os nomes de identificadores devem ser constituídos apenas de letras, números e sublinhados (_)
- Os nomes de identificadores não podem começar com números
- Os nomes de identificadores não podem ser uma palavra reservada da linguagem

**Observação:** Go é uma linguagem Case Sensitive, então ela diferencia caracteres de maiúsculas para minúsculas

## 2.1 Comentários

Comentários em Go são linhas que são marcadas para não serem executadas durante a compilação, linhas que serão ignoradas. Podem ser usadas para marcações e anotações durante o código.

Os comentários são formados das seguintes formas:

- Comentários de linha única: // Aqui será digitado o comentário a não ser executado
- Comentários de linhas múltiplas: /* Aqui será digitado o comentário delimitado pela barra e asterisco * /

## 2.2 Palavras Reservadas

Em programação, palavras-chave, ou palavras reservadas, são as palavras que não podem ser usadas como identificadores. Em outras palavras, não podem ser usadas como nome de variáveis, nome de classes, etc. Estas palavras são assim definidas ou porque já têm uso na sintaxe da linguagem ou porque serão usadas em alguns momentos, seja para manter compatibilidade com versões anteriores ou mesmo com outras linguagens. No caso do Go temos as seguintes palavras reservadas:

**float** - usada para declarar uma variável que pode conter um número de ponto flutuante de 32 bits.

**int** - usada para declarar uma variável que pode conter um número do tipo inteiro 32 bits.

**bool** - usada para declarar uma variável como um tipo boolean. Pode conter apenas valores true e false.

**string** - são sequências de caracteres entre sinais de crase, frequentemente chamadas de backticks.

**break** - “quebra” um comando de repetição (for), encerrando o bloco antes da condição ser falsa.

**func** - dá suporte a abstrações de funções.

**interface** - a palavra-chave interface é usada para declarar uma interface. Pode ter apenas métodos abstratos.

**case** - a palavra-chave case é usada com as instruções switch para marcar blocos de texto.

**go** - a linguagem Go possui nativamente uma série de comandos operacionais. Você pode executar o comando go na sua linha de comando para ver a lista completa.

**map** - é uma coleção de pares chave-valor, sem nenhuma ordem definida.

**struct** - define um tipo struct no código.

**chan** - channels são uma espécie de túnel de comunicação entre goroutines, onde uma goroutine consegue enviar informações para outra antes mesmo de terminar sua execução.

**else** - utilizada para indicar as ramificações alternativas em uma instrução if.

**package** - utilizada para declarar um pacote Go que inclui as classes.

**switch** - a palavra-chave switch contém uma instrução switch que executa o código com base no valor do teste. A instrução switch testa a igualdade de uma variável em relação a vários valores.

**const** - define uma variável como constante, ela não poderá ser alterada no decorrer do código.

**fallthrough** - nos permite executar o próximo bloco de case sem verificar sua condição. Em outras palavras, podemos mesclar dois blocos case usando a palavra-chave fallthrough.

**if** - testa a condição. Executa o bloco if se a condição for verdadeira.

**range** - itera com elementos em uma variedade de estruturas de dados.

**type** - retorna o tipo de uma variável passada como parâmetro.

**continue** - a instrução continue dá a opção de ignorar a parte de um loop onde uma condição externa é acionada, mas continuar e completar o resto do loop.

**for** - Tem o objetivo de executar um conjunto de instruções repetidamente enquanto suas condições sejam verdadeiras.

**import** - torna as classes e interfaces disponíveis e acessíveis ao código-fonte atual.

**return** - usada para retornar um método quando sua execução estiver concluída.

**var** - usada para declarar uma variável.

## 2.3 Literais reservados

**nil** - Equivalente a null em outras linguagens.

**true** - valor Booleano.

**false** - valor Booleano.

## 2.4 Operadores e Delimitadores

| **Tipos de operadores** | **Categoria** | **Precedência** |
|---|---|---|
| Unários | Prefixo | ++exp - -exp |
| | Sufixo | exp++ exp - - |
| Aritméticos | Multiplicação | * |
| | Soma | + |
| | Subtração | - |
| | Divisão | / % |
| Relacionais | Igual | == |
| | Diferente de | != |
| | Maior que | > |
| | Menor que | < |
| | Maior igual | >= |
| | Menor igual | <= |
| Lógicos | Conjunção (E) | && |

| | | |
|---|---|---|
| | Disjunção (Ou) | \| \| |
| | Negação | != |
| Atribuição | Atribuições em Go separados por \ | = \ += \ -= \ *= \ /= \ %= \ := |

### 2.5 Literais String

Uma string é uma sequência de um ou mais caracteres que podem estar em uma constante ou uma variável. Criadas com o padrão de codificação de caracteres Unicode, as strings são sequências imutáveis, o que significa que elas não podem ser modificadas.

**Ex:** variavel := “String que será atribuída"

```
fmt.Println(variavel)
```

#### 2.5.1 Strings com caracteres UTF-8

O UTF-8 é um esquema de codificação usado para codificar caracteres de largura variável em um a quatro bytes. A linguagem Go já vem totalmente compatível com os caracteres UTF-8, sem qualquer configuração especial, bibliotecas ou pacotes.

**Ex:** a := `"Hello, 世界"`

### 2.6 Literais Numéricos

#### 2.6.1 Literais Inteiros

Os tipos inteiros de Go são: **uint8, uint16, uint32, uint64, int8, int16, int32 e int64.** 8, 16, 32 e 64 nos dizem quantos bits cada um dos tipos usa. **uint** significa “inteiro sem sinal” enquanto int significa “inteiro com sinal”. Inteiros sem sinal contém apenas números positivos (ou zero).

#### 2.6.2 Literais de ponto flutuante

Um número de ponto flutuante ou float é usado para representar números reais que não podem ser expressos como inteiros. Os números reais incluem todos os números

racionais e irracionais e, por isso, os números de ponto flutuante podem conter uma parte fracionada.

**Ex:** variavel := 3.14

### 2.6.3 Literais Booleanos

Literais booleanas são representações de código para os valores booleanos. Sendo assim, seus valores só poderão ser **true** ou **false**.

**Ex:** variavel := true **//Essa variável servirá como um valor verdadeiro para testes e expressões**

**Referência:**


SILVA, Gizele. Go (GoLang): conheça a linguagem criada pelo Google. 2011. Disponível em: https://coodesh.com/blog/candidates/carreiras/go-lang-developers/ .

GUIDES, Gopher. Como usar variáveis e constantes em Go. 13 feb 2020. Disponível em: https://www.digitalocean.com/community/tutorials/how-to-use-variables-and-constants-in-go-pt .

GUIDES, Gopher. Entendendo os tipos de dados em Go. 13 feb 2020. Disponível em: https://www.digitalocean.com/community/tutorials/understanding-data-types-in-go-pt .

DOXSEY, Caleb. An introduction to programming in Go. 2012. Disponível em: https://www.golang-book.com/books/intro .

CARUSO, Guilherme. GoSchool: Operadores. 15 nov 2018. Disponível em: https://medium.com/gommunity/operadores-c7e1a41cfd4 .

Um tutorial sobre a linguagem Go. [S. l], 2014. Disponível em: http://www.inf.ufes.br/~vitorsouza/archive/2020/wp-content/uploads/teaching-lp-20142-seminario-go.pdf .