

Sessão 3 - Listas, Arrays, Slices e Maps em Go

18. Conceito de Lista:

Listas em Go são implementadas como slices.

Exemplo:

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    var lista []int
```

```
    fmt.Println("Lista vazia:", lista)
```

```
}
```

19. Criando Listas:

Slices podem ser criados diretamente ou com a função `make`.

Exemplo:

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
lista := []int{1, 2, 3, 4, 5}

fmt.Println("Lista:", lista)

}
```

20. Make e For para Listas:

A função `make` cria slices com tamanho e capacidade inicial.

Exemplo:

```
package main

import "fmt"

func main() {

    lista := make([]int, 5)

    for i := 0; i < len(lista); i++ {

        lista[i] = i + 1

    }

    fmt.Println("Lista criada com make:", lista)

}
```

21. Sublistas:

Obter uma sublista de um slice original.

Exemplo:

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    lista := []int{1, 2, 3, 4, 5}
```

```
    sublista := lista[1:4]
```

```
    fmt.Println("Sublista:", sublista)
```

```
}
```

22. Index:

Acessar elementos de um slice com índice.

Exemplo:

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    lista := []int{10, 20, 30, 40, 50}
```

```
    fmt.Println("Elemento no índice 2:", lista[2])
```

```
}
```

23. Array vs Slices:

Arrays têm tamanho fixo, slices são dinâmicos.

Exemplo:

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    var array [5]int
```

```
    fmt.Println("Array:", array)
```

```
    slice := []int{1, 2, 3}
```

```
    fmt.Println("Slice:", slice)
```

```
}
```

24. Maps:

Estrutura de dados que associa chaves a valores.

Exemplo:

```
package main
```

```
import "fmt"
```

```
func main() {  
    pessoas := map[string]int{  
        "Alice": 30,  
        "Bob": 25,  
    }  
  
    fmt.Println("Idade de Alice:", pessoas["Alice"])  
}
```

25. Maps na Prática:

Adicionar, modificar e excluir elementos de um map.

Exemplo:

```
package main
```

```
import "fmt"
```

```
func main() {  
    pessoas := map[string]int{}  
    pessoas["Lincoln"] = 29  
    pessoas["Maria"] = 22  
  
    fmt.Println("Idades:", pessoas)  
  
    pessoas["Lincoln"] = 30
```

```
    fmt.Println("Nova idade de Lincoln:", pessoas["Lincoln"])
}
```

26. Range:

Iterar sobre slices, arrays ou maps usando `range`.

Exemplo:

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    lista := []int{1, 2, 3, 4, 5}
```

```
    for i, valor := range lista {
```

```
        fmt.Println("Índice:", i, "Valor:", valor)
```

```
    }
```

```
}
```

27. Delete:

Remover elementos de um map com `delete`.

Exemplo:

```
package main
```

```
import "fmt"

func main() {
    pessoas := map[string]int{"Lincoln": 29, "Maria": 22}
    fmt.Println("Antes de deletar:", pessoas)

    delete(pessoas, "Lincoln")
    fmt.Println("Depois de deletar Lincoln:", pessoas)
}
```