01

# Arrays em Go

**Transcrição**

Neste capítulo, vamos evoluir o monitoramento do nosso programa, monitorando mais de um site ao mesmo tempo. Como programadores, não vamos criar uma variável para cada site que desejamos monitorar, e sim uma estrutura de dados responsável por conter várias strings, vários sites, o já conhecido **Array**.

Então, vamos ver como trabalhamos com coleções, principalmente Arrays e Slices, na linguagem Go.

## Declarando um array

Primeiramente, vamos criar um array com a estrutura clássica, com o `var`. Para isso, além do `var`, nós devemos informar o nome do array, colchetes e o tipo de dados que ele guardará. Além disso, dentro dos colchetes, devemos informar o tamanho do array:

```
// restante do código omitido

func iniciarMonitoramento() {
    fmt.Println("Monitorando...")
    var sites [4]string
    site := "https://www.alura.com.br"
    resp, _ := http.Get(site)

    if resp.StatusCode == 200 {
        fmt.Println("Site:", site, "foi carregado com sucesso!")
    } else {
        fmt.Println("Site:", site, "está com problemas. Status Code:", resp.StatusCode)
    }
}
```

É importante nos atentar ao tipo de dados do array, não há espaço entre ele e os colchetes, na hora da declaração do array.

## Colocando um valor dentro do array

Para colocar um valor no array, não há mistério, basta atribuir um valor a algum dos seus índices:

```
// restante do código omitido

func iniciarMonitoramento() {
    fmt.Println("Monitorando...")

    var sites [4]string
    sites[0] = "https://random-status-code.herokuapp.com/"
    sites[1] = "https://www.alura.com.br"
    sites[2] = "https://www.caelum.com.br"

    site := "https://www.alura.com.br"
    resp, _ := http.Get(site)
```

```
        if resp.StatusCode == 200 {
            fmt.Println("Site:", site, "foi carregado com sucesso!")
        } else {
            fmt.Println("Site:", site, "está com problemas. Status Code:", resp.StatusCode)
        }
    }
```

Mas o fato do array ter um tamanho fixo, nos limita um pouco, pois se quisermos adicionar 5 itens no array, teremos que alterar o seu tamanho na sua declaração. Por isso, em Go, geralmente não trabalhamos com arrays, e sim com uma outra estrutura de dados, chamada **Slice**, que funciona em cima do array, mas não tem tamanho fixo.

Veremos mais sobre isso no próximo vídeo.