

Mãos na Massa: Delay e Aumentando o número de testes

Agora vamos fazer com que a cada vez que o usuário selecione a opção de monitoramento, cada site seja testado mais de uma vez, de acordo com o que o usuário setar nas constantes.

1- Primeiro crie a constante `monitoramentos` que indicará quantas vezes o site será testado. Coloque o número que desejar, mas lembre-se que quanto mais você testar, mais o script vai demorar:

```
//hello.go
package main

import (
    "fmt"
    "net/http"
    "os"
)

const monitoramentos = 3

// restante do arquivo
```

2- Agora vamos colocar um `for` dentro da função `iniciarMonitoramentos` para que o slice seja percorrido o número de vezes que configuramos na constante:

```
//hello.go

//restante do arquivo

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

    sites := []string{"https://random-status-code.herokuapp.com/",
        "https://www.alura.com.br", "https://www.caelum.com.br"}

    //Adição aqui
    for i := 0; i < monitoramentos; i++ {
        for i, site := range sites {
            fmt.Println("Testando site", i, ":", site)
            testaSite(site)
        }
    }

    fmt.Println("")
}
```

3- Para que haja uma pausa entre cada um dos testes que faremos, vamos adicionar um pequeno `delay` entre cada vez cada iteração de monitoramento. Utilize a função `sleep` do pacote `time` para dar uma pausa entre monitoramentos. Para que o tamanho desta pausa fique configurável, vamos criar uma constante que vai dizer o tamanho do delay:

```
//hello.go
package main

import (
    "fmt"
    "net/http"
    "os"
)

const monitoramentos = 3
const delay = 5

// restante do arquivo
```

4- E a partir do valor da constante, vamos adicionar um `time.Sleep` na função `iniciarMonitoramento`. Vamos multiplicar a nossa contante (`delay`) pela constante que representa o número de segundos no Go (`time.Second`):

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

    sites := []string{"https://random-status-code.herokuapp.com/",
        "https://www.alura.com.br", "https://www.caelum.com.br"}

    for i := 0; i < monitoramentos; i++ {
        for i, site := range sites {
            fmt.Println("Testando site", i, ":", site)
            testaSite(site)
        }

        // adição AQUI!
        time.Sleep(delay * time.Second)
        fmt.Println("")
    }
    fmt.Println("")
}
```

Adicione o outro `fmt.Println("")` para que a exibição fique mais organizada para o seu usuário

Agora o seu script deve estar testando os seus sites de acordo com o número de vezes que você setou na constante `monitoramentos`, e entre cada testada ele deve dar uma pausa de acordo com o número de segundos que você configurou na constante `delay`.