



Programação em Go

▶ Ponteiros e Canais em Go

// Exercícios 3 - Go Bases

Objetivo

O objetivo deste exercício é poder consolidar os conceitos sobre funções, vistos no módulo Go Bases. Para isso vamos propor uma série de exercícios simples e incrementais (já que vamos trabalhar e adicionar complexidade ao que temos que construir), que nos permitirão rever os tópicos que estudamos.

Fluxo de trabalho

Para resolver os exercícios, eles devem ser feitos em seus computadores. Lembramos você de criar uma pasta para cada aula e ter um arquivo .go dentro de cada exercício.

Are you ready? 😊👍



Exercício 1 - Rede social

Uma empresa de mídia social precisa implementar uma estrutura de usuários com funções que acrescentem informações à estrutura. Para otimizar e economizar memória, eles exigem que a estrutura de usuários ocupe o mesmo lugar na memória para o programa principal e para as funções:

- A estrutura deve possuir os seguintes campos: Nome, Sobrenome, idade, email e senha

E devem implementar as seguintes funcionalidades:

- mudar o nome: me permite mudar o nome e sobrenome
- mudar a idade: me permite mudar a idade
- mudar e-mail: me permite mudar o e-mail
- mudar a senha: me permite mudar a senha



Exercício 2 - E-commerce (Parte II)

Uma grande empresa de vendas na web precisa adicionar funcionalidades para adicionar produtos aos usuários. Para fazer isso, eles exigem que usuários e produtos tenham o mesmo endereço de memória no main do programa e nas funções.

Estruturas necessárias:

- Usuário: Nome, Sobrenome, E-mail, Produtos (array de produtos).
- Produto: Nome, preço, quantidade.

Algumas funções necessárias:

- Novo produto: recebe nome e preço, e retorna um produto.
- Adicionar produto: recebe usuário, produto e quantidade, não retorna nada, adiciona o produto ao usuário.
- Deletar produtos: recebe um usuário, apaga os produtos do usuário.



Exercício 3 - Calcular Preço (Part II)

Uma empresa nacional é responsável pela venda de produtos, serviços e manutenção.

Para isso, eles precisam realizar um programa que seja responsável por calcular o preço total dos Produtos, Serviços e Manutenção. Devido à forte demanda e para otimizar a velocidade, eles exigem que o cálculo da soma seja realizado em paralelo por meio de 3 go routines.

Precisamos de 3 estruturas:

- Produtos: nome, preço, quantidade.
- Serviços: nome, preço, minutos trabalhados.
- Manutenção: nome, preço.

Precisamos de 3 funções:

- Somar Produtos: recebe um array de produto e devolve o preço total (preço * quantidade).
- Somar Serviços: recebe uma array de serviço e devolve o preço total (preço * média hora trabalhada, se ele não vier trabalhar por 30 minutos, ele será cobrado como se tivesse trabalhado meia hora).
- Somar Manutenção: recebe um array de manutenção e devolve o preço total.

Os 3 devem ser executados concomitantemente e ao final o valor final deve ser mostrado na tela (somando o total dos 3).



Exercício 4 - Ordenamento

Uma empresa de sistemas precisa analisar que algoritmos de ordenamento utilizar para seus serviços.

Para eles é necessário instanciar 3 arrays com valores aleatórios não ordenados

- uma matriz de inteiros com 100 valores
- uma matriz de inteiros com 1000 valores
- uma matriz de inteiros com 10.000 valores

Para instanciar as variáveis, utilize o rand:

```
package main

import (
    "math/rand"
)

func main() {
    variavel := rand.Perm(100)
    variave2 := rand.Perm(1000)
    variave3 := rand.Perm(10000)
}
```

Cada um deve ser ordenado por:

- Inserção
- grupo
- seleção

Uma rotina para cada execução de classificação

Tenho que esperar terminar a ordenação de 100 números para continuar com a ordenação de 1000 e depois a ordenação de 10000.

Por fim, devo medir o tempo de cada um e mostrar o resultado na tela, para saber qual ordenação ficou melhor para cada arranjo.