**Readme**

# Explicação da minha resolução diante do DesafioTécnico - Estágio/Junior Go(lang)

O desafio propôs o seguinte:

Imagine uma lista de compras. Ela possui:

* Itens
* Quantidade de cada item
* Preço por unidade/peso/pacote de cada item

Desenvolva uma função (ou método) que irá receber uma lista de compras (como a detalhada acima) e uma lista de e-mails. Aqui, cada e-mail representa uma pessoa.

A função deve:

* Calcular a soma dos valores, ou seja, multiplicar o preço de cada item por sua quantidade e somar todos os itens
* Dividir o valor de forma igual entre a quantidade de e-mails
* Retornar um mapa/dicionário onde a chave será o e-mail e o valor será quanto ele deve pagar nessa conta.

1) Minha Ideia

Assim sendo, eu cheguei no seguinte raciocínio: Eu tenho 2 listas e tenho que transformar isso em um dicionário/mapa. Uma lista é de E-mails e representa um indivíduo, a outra lista é de Itens, cada Item possuí um nome, quantidade e seu respectivo valor por unidade. Eu vou ter que fazer um cálculo e somar o valor total de todos os itens e, a partir desse cálculo, eu já tenho todos os atributos necessários para formar o mapa. Eu vou ter um **ValorTotal** e o tamanho da lista de E-mails, com isso eu posso calcular o **ValorIndividual** que cada pessoa deverá pagar, porém existem algumas ressalvas importantes:

O **ValorIndividual** pode ser calculado através da seguinte expressão --> **ValorIndividual = ValorTotal / tamanho da lista de E-mails.** Aqui existem duas possibilidades, uma divisão exata (com resto = 0) e uma divisão não exata (com resto != 0).

Se a divisão for exata o mapeamento já pode ser feito diretamente, usando-se o **E-mail** como **chave** e o **ValorIndividual** como **Valor**. A estrutura fica assim:  
Mapa[**E-mail**] = **ValorIndividual.**

Se a divisão não for exata eu vou ter um tratamento um pouco diferente, teremos que distribuir o resto da divisão para as respectivas chaves visando minimizar a diferença do **ValorIndividual** entre cada pessoa**.** A estrutura fica assim:

Mapa[**E-mail**] = **ValorIndividual + uma parte do resto**.

2) Como eu implemento essa Ideia?

Imagine 2 listas, uma de E-mails e outra de Itens, da seguinte forma:

ListaDeEmails = [“João@hotmail.com”, “Zé@hotmail.com”, [Paulo@hotmail.com](mailto:Paulo@hotmail.com)]

ListaDeItens = [{“Arroz”, 2(quantidaDeItem), 200(valorItem)}, {“Feijão”, 4, 100}]

1°) Calcular o **ValorTotal**:

* Eu vou fazer uma iteração na lista de Itens (do início ao fim). Em cada iteração eu vou atribuir ao **ValorTotal =** ValorTotal + quantidaDeItem \* valorItem.

2°) Calcular o **ValorIndividual** e o **resto**:

- Serão calculados através das seguintes expressões:

**ValorIndividual** = ValorTotal / len(ListaDeEmails).

**Resto** = ValorTotal % len(ListaDeEmails)

-É importante ressaltar que aqui existe a possiblidade de ocorrer uma exceção, isto é, quando len(ListaDeEmails) == 0. Se isso acontecer não tem como fazer o mapeamento, pois resulta em um infinito (divisão por 0).

3°) Atribuir o **ValorIndividual** para as **chaves(E-mail)** do dicionário:

- Eu vou fazer uma iteração na lista de E-mails (do início ao fim), em cada iteração eu vou atribuir ao mapa o respectivo **E-mail** como **chave** e o **ValorIndividual** como **valor.**

4°) Analisar o **resto**:

- Se o resto for igual a 0, quer dizer que o meu mapa já está formado, e a atribuição já está correta, logo eu retorno o meu mapa.

- Se o resto for diferente de 0, eu vou fazer uma nova iteração na lista de E-mails (no intervalo do início até o resto) incrementando em uma unidade (1) o **ValorIndividual** da respectiva **chave.** E por fim eu retorno o novo mapa com o resto distribuído.

3) Como eu cheguei nessa solução e minha experiência com esse desafio

No meu ponto de vista, a parte mais importante para se chegar nessa solução, é com base no propósito do que foi descrito no tópico anterior, o da implementação. A implementação precisa da ideia, e depois de ler o enunciado do desafio, eu tive algumas ideias para resolvê-lo, porém eu não tinha os recursos para sustentar e desenvolver essa ideia, neste caso, o conhecimento específico da linguagem Go. A partir daí eu desenvolvi essa ideia utilizando os recursos que estavam ao meu alcance, no caso, a linguagem Python. Deu certo. Agora o meu desafio era “traduzir” o código Python para o Golang. Assisti alguns vídeos de tutoriais sobre o Go para começar a pegar intimidade, depois, aos poucos, eu fui dando pequenos passos para atender o meu objetivo final. Python é bem diferente do Go, eu tive que criar meu próprio tipo, através de uma Struct. Eu tive que aprender a criar uma Slice e iterar sobre a mesma. Eu tive que aprender a adicionar um novo item em uma Slice. Eu tive que aprender a passar um argumento pra minha função e retornar um determinado tipo da mesma. No final das contas eu fiquei muito feliz por ter conseguido alcançar o objetivo que, resumidamente, seria: pegar duas listas e retornar um mapa. Eu tenho consciência que essa não é a melhor solução, mas não deixa de ser uma solução. Eu fiz tudo sozinho com muita pesquisa na Internet e, aos poucos, eu vou melhorando e me tornando mais capaz, esse foi só o pontapé inicial !!

4) Materiais que usei como referência:

<https://gobyexample.com/>

<https://www.youtube.com/c/AprendaGo/playlists>

<https://www.geeksforgeeks.org/how-to-pass-a-slice-to-function-in-golang/>

<https://blog.golang.org/maps>