**Desafio Técnico**

A ideia deste desafio é resolver um problema comum no dia-a-dia de quem divide almoços/lanches com a equipe de trabalho. Vamos imaginar que você e mais um colega queiram dividir um lanche que estão pedindo pelo iFood/Uber Eats utilizando seu smartphone. Fica fácil descobrir quanto seu colega deverá lhe pagar quando não existe nenhum desconto ou valor de entrega, porém quando estes fatores entram em questão, simplesmente dividir o valor no meio pode não ser o mais justo.

Para ilustrar melhor esta situação, vamos supor que você pediu um hamburguer de R$40,00 e uma sobremesa de R$2,00, enquanto seu amigo pediu apenas um sanduíche de R$8,00. Logo, você pagará inicialmente um total de R$42,00 enquanto seu amigo pagará R$8,00. Porém, considere também que o pedido total teve um valor adicional de R$8,00 pela entrega e que houve um desconto de R$20,00 no total do pedido. Sendo assim, é justo que a entrega e o desconto sejam proporcionais ao valor dos itens que cada um comprou. Neste caso, temos o seguinte:

* Hamburguer = 40,00
* Sobremesa = 2,00
* Sanduíche (amigo) = 8,00
* Total = 50,00
* Entrega = 8,00
* Desconto = 20,00
* Total a pagar = 38,00

Logo, dos R$38,00, o seu amigo deverá pagar R$6,08, enquanto você pagará R$31,92, que corresponde ao desconto e entrega proporcionais aos itens que foram solicitados por cada um.

Com o objetivo de facilitar esta conta, você deverá criar uma pequena aplicação que irá calcular o total que seus amigos deverão lhe pagar quando dividirem um almoço. Ao final, você deverá simplificar o pagamento gerando um link do Picpay (ou outra carteira de sua preferência), para enviar a cobrança a seus amigos. \*Obs.: Podem existir inúmeras formas de facilitar esta cobrança, e o Picpay aqui mencionado é apenas uma delas, portanto deixe seu código adaptável para receber quaisquer outras formas de pagamento que poderão ser posteriormente integráveis à sua aplicação.

## Requisitos necessários

1. Back-end deve ser feito em Java, utilizando frameworks de sua escolha
2. Deverão haver testes unitários
3. Considerar que você pode dividir a conta com um ou mais amigos
4. Considerar que podem haver descontos e acréscimos em reais, e também em porcentagem (por exemplo os 10% do garçom, quando o almoço for em um restaurante físico)
5. Front-end não será avaliado, já que o foco deste desafio é o back-end, porém deverá conter o mínimo necessário para inputar os dados e interagir com o back-end
6. Back/front devem estar totalmente desacoplados
7. Código deve ser disponibilizado em um repositório git (bitbucket, github e afins)

## O que será avaliado

1. Clean-code
2. Organização do código e uso de Orientação a Objetos
3. Manutenibilidade
4. Testes