

Cases

Vamos explorar alguns outros cenários, com fluxo condicionais, repetições e excepcionais.

Case 1: Vamos imaginar que em um processo seletivo, existe o valor base salarial de R\$ 2.000,00 e o salário pretendido pelo candidato. Vamos elaborar um controle de fluxo onde:

- Se o valor salário base, for maior que valor salário pretendido, imprima : **LIGAR PARA O CANDIDATO;**
- Senão, Se o valor salário base for igual ao valor salário pretendido, imprima : **LIGAR PARA O CANDIDATO, COM CONTRA PROPOSTA;**
- Senão imprima: **AGUARDANDO RESULTADO DOS DEMAIS CANDIDATOS.**

Case 2: Foi solicitado, que nosso sistema garanta que, diante das inúmeras candidaturas sejam selecionados apenas no máximo, 5 candidatos para entrevista, onde o salário pretendido seja menor ou igual ao salário base.

```
// Array com a lista de candidatos
```

```
String [] candidatos = {"FELIPE", "MÁRCIA", "JULIA", "PAULO", "AUGUSTO", "MÔNICA", "FABRÍCIO", "
```

```
// Método que simula o valor pretendido
```

```
import java.util.concurrent.ThreadLocalRandom;  
static double valorPretendido() {  
    return ThreadLocalRandom.current().nextDouble(1800, 2200);  
}
```

Case 3: Agora é hora de imprimir a lista dos candidatos selecionados, para disponibilizar para o RH entrar em contato.

Case 4: O RH deverá realizar uma ligação, com no máximo 03 tentativas para cada candidato selecionado e caso o candidato atenda, deve-se imprimir:

- **"CONSEGUIMOS CONTATO COM _ [CANDIDATO] ' ' ' APÓS **_** [TENTATIVA] '** TENTATIVA(S)" ;****
- Do contrário imprima: **"NÃO CONSEGUIMOS CONTATO COM O _ [CANDIDATO] _".**

Soluções

[Case 1](#)[Case 2](#)[Case 3](#)[Case 4](#)

Hora de aplicar os conceitos apresentados, implementando as soluções, para cada case apresentado.

[Previous](#)
[Estruturas excepcionais](#)

[Next](#)
[Conceito de POO](#)

Last updated 1 year ago