



## 06. Finally e try with resources

📅 Date	@30/09/2022
▼ Categoria	Java
▼ Curso	Java Exceções: aprenda a criar lançar e controlar exceções

### Tópicos

- Finally
- Palavras chaves
- Try with resources
- Bloco finally
- Tratamento de exceções
- try com recurso
- Mãos na massa: Finally e try with resources
- Para saber mais: Exceções padrões
- O que aprendemos?
- Conclusão
- Projeto final
- Arquivos do projeto atual

### Para saber mais: Exceções padrões

No vídeo, usamos a exceção `IllegalStateException`, que faz parte da biblioteca do Java e indica que um objeto possui um estado inválido. Você já deve ter visto outras exceções famosas, como a `NullPointerException`. Ambos fazem parte das exceções padrões do mundo Java que o desenvolvedor precisa conhecer.

Existe outra exceção padrão importante que poderíamos ter usado no nosso projeto. Para contextualizar, faz sentido criar uma conta com uma agência que possui valor negativo? Por exemplo:

```
Conta c =newContaCorrente(-111, 222); //faz sentido?COPIAR CÓDIGO
```

Não faz sentido nenhum. Por isso, poderíamos verificar os valores no construtor da classe. Caso o valor esteja errado, lançamos uma exceção. Qual?

A `IllegalArgumentException`, que faz parte das exceções do biblioteca do Java:

```
publicabstractclassConta {  
  
    //atributos omitidos  
    public Conta(int agencia, int numero){  
        if(agencia < 1) {  
            throw new IllegalArgumentException("Agencia inválida");  
        }  
        if(numero < 1) {  
            throw new IllegalArgumentException("Numero da conta inválido");  
        }  
        //resto do construtor foi omitido  
    }  
}
```

A `IllegalArgumentException` e `IllegalStateException` são duas exceções importantes, que o desenvolvedor Java deve usar. Em geral, quando faz sentido, use uma exceção padrão em vez de criar uma própria.

## O que aprendemos?

Nessa aula, aprendemos:

- que existe um bloco `finally`, útil para o fechamento de recursos (como conexão);

- quando há um bloco `finally` o bloco `catch` é opcional;
  - que o bloco `finally` é sempre executado, sem ou com exceção;
  - como usar o *try-with-resources*.
-