

# PROJETO, IMPLEMENTAÇÃO E TESTE DE SOFTWARE

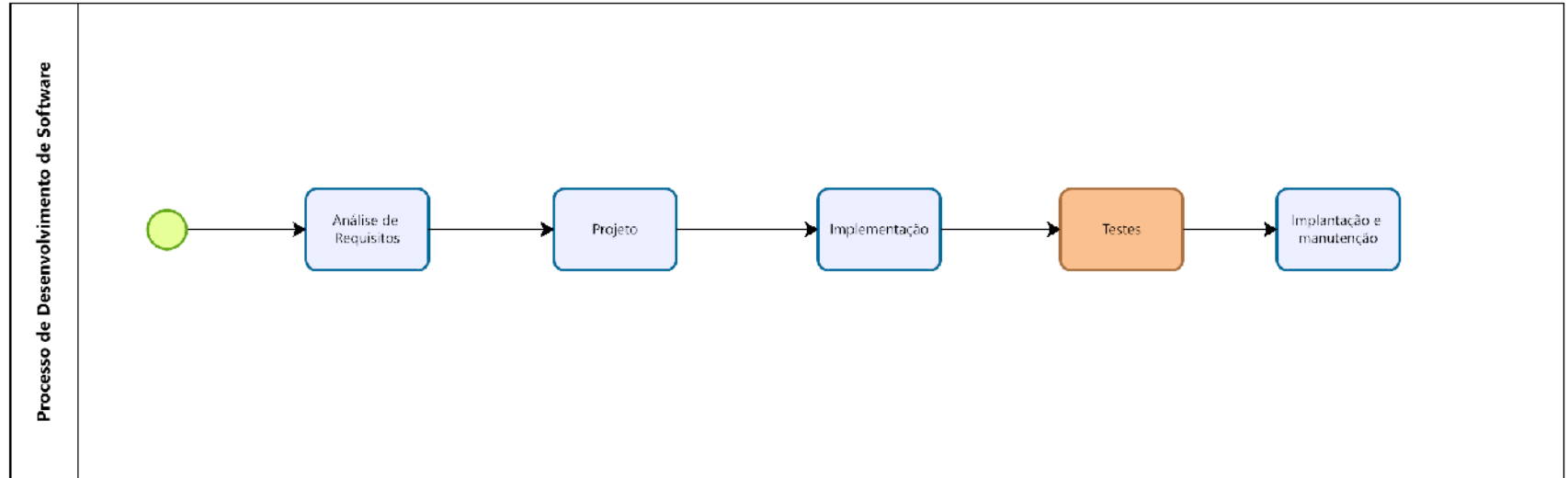
## Aula 08 – TESTE DRIVEN DEVELOPMENT

Professor Fabricio Freire

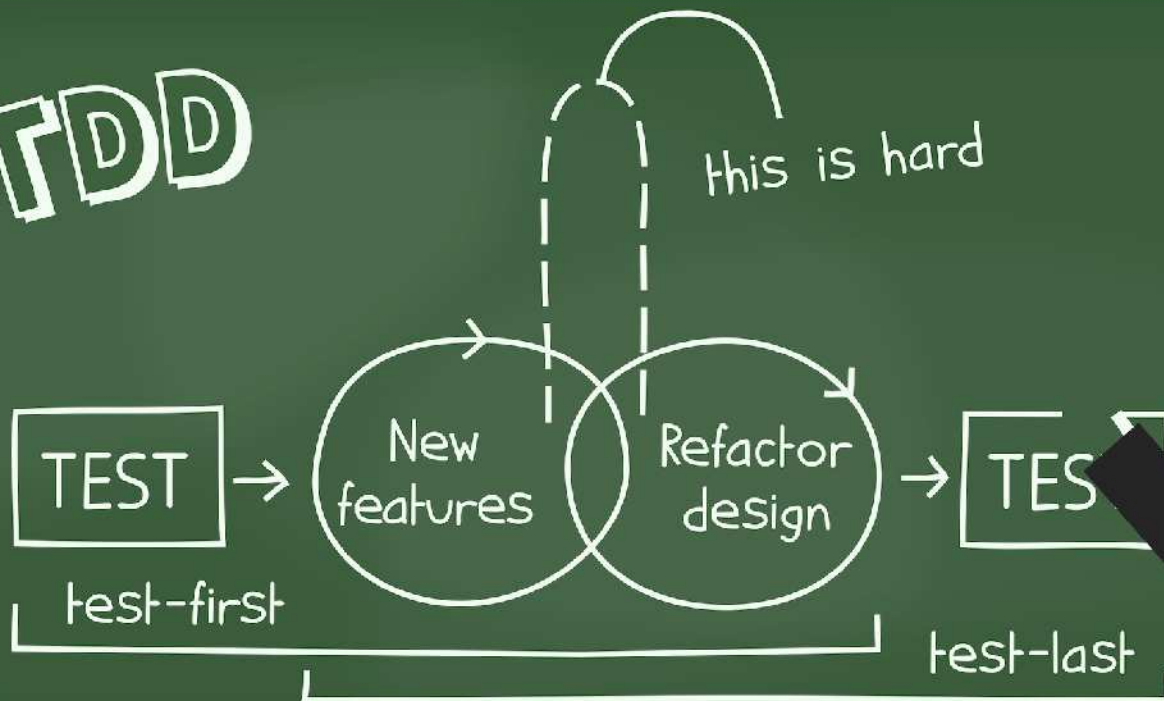
# METAS DE APRENDIZAGEM

- Introdução
- Etapas
- F.I.R.S.T.
- Benefícios
- Implementação

## ONDE ESTAMOS



**TDD**



# O QUE É TDD?

- Abreviatura de Test Driven Development;
- Inversão no fluxo de desenvolvimento;
- Surgiu nos anos 90, e está associado a metodologias ágeis, como o XP;
- Fomento à simplicidade.

# ETAPAS DO TDD

## Red

- Código para uma nova funcionalidade;
- O intuito é escrever um código que irá falhar;
- Maior compreensão dos requisitos.

## Green

- Melhoria no código escrito na etapa Red – Foco!!!!
- Implemente o mínimo possível;
- Código deve passar nos testes.

## Refactor

Melhoria no código que passou nos testes – tornar mais legível

# F.I.R.S.T

Acrônimo para:

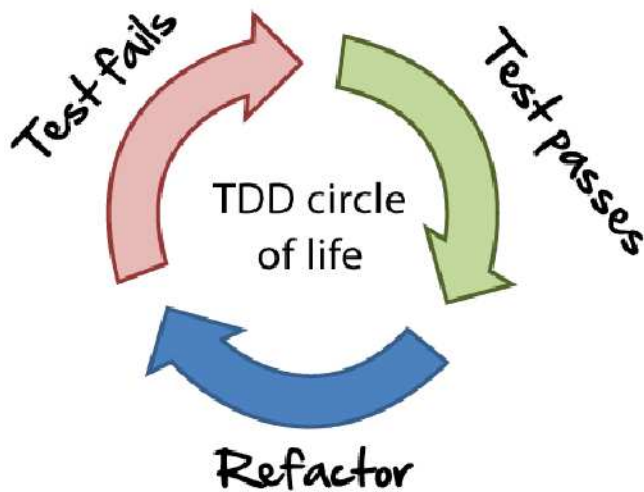
**F (Fast)** – Rápidos

**I (Isolated)** – Isolados

**R (Repeatable)** – Reproduzíveis

**S (Self-verifying)** – Auto verificáveis

**T (Timely)** - Oportunos



# BENEFÍCIOS

**Melhor qualidade** – Código testado de forma abrangente

**Melhor documentação** – Os teste podem servir como documentação

**Menor risco de regressão** – Identificação antecipada de problemas

**Maior produtividade** – Elimina ou diminui o tempo gasto em testes manuais



# IMPLEMENTAÇÃO

Utilização das ferramentas:

Java

Spring Boot

JUnit

Mockito



Spring Boot



JUnit



# IMPLEMENTAÇÃO

Vamos à prática

**“Ao considerar que o desenvolvedor comete erros banais muitas vezes deixamos de entrar fundo nos seus testes, que também acabam sendo negligenciados.”**

**BONS ESTUDOS**