

# **Testes automatizados**

**Prof. Dr. Bruno Queiroz Pinto**

# Currículo

- Graduação em Ciência da Computação pela Universidade de Marília, 2000;
- Mestrado em Ciência da Computação pela Universidade Federal de Uberlândia, 2003;
- Doutor em Ciência da Computação pela Universidade Federal Fluminense.

The image shows a blank curriculum vitae form titled "CURRICULUM VITAE". The form is divided into several sections:

- DADOS PESSOAIS** (Personal Data): Includes fields for Name, Surname, Date of Birth, Place of Birth, Marital Status, and Address.
- ESCOLARIDADE** (Education): Includes sections for "CURSO 1º GRAU" (First Degree), "CURSO 2º GRAU" (Second Degree), and "CURSO SUPERIOR" (Higher Education), each with fields for Course, Institution, Start Date, End Date, and Grade.
- OUTROS DADOS** (Other Data): Includes fields for Telephone, E-mail, and a section for "Outros Dados" (Other Data).

The form is designed to be filled out by the individual, providing a structured overview of their educational and personal background.

# Disciplinas 2026/1

- Disciplinas :
  - Projeto e Desenvolvimento de Sistemas (5° LIC)
  - Lógica de Programação (1° TSPI)
  - Testes automatizados (3° TSPI)

# Contatos

- Email:

[bruno.queiroz@iftm.edu.br](mailto:bruno.queiroz@iftm.edu.br)

- Repositório com os projetos desenvolvidos

[https://github.com/brunoqp78/testes\\_automatizados\\_2026-1](https://github.com/brunoqp78/testes_automatizados_2026-1)

## Dados da disciplina (I)

- **Curso:** Tecnologia em Sistemas para Internet
- **Carga Horária:** 83
- **Horários: Segunda-feira** (4 horários + 1 horário em EAD)

## Dados da disciplina (II)

- **Objetivos:**

- Criação de testes de unidade, para que cada unidade do sistema seja testada individualmente.

JUnit

- Construção de testes de integração, para que módulos completos do sistema sejam testados.

JUnit + Mockito e MockMVC

- Construção de testes de aceitação para que as interfaces do sistema sejam testadas automaticamente.

Selenium

- Utilização de métricas para aferir cobertura de testes.

Pesquisa + JaCoCo

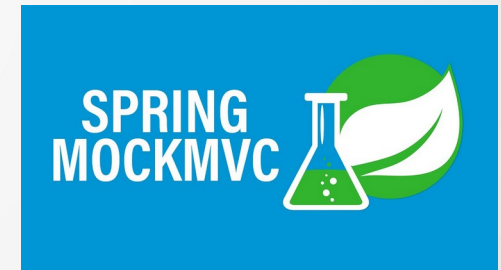
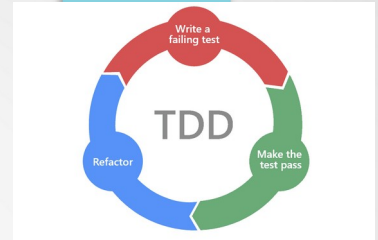
# Conteúdo Programático (I)

1. Fundamentos de Testes automatizados
  - Conceitos Básicos
  - Boas práticas e padrões
2. Testes de Unidade Vanilla - JUnit
  - Não utilizar a estrutura do framework Spring Boot



# Conteúdo Programático (I)

3. TDD - Test Driven Development
4. Dublês de Teste e Isolamento (Mockito)
5. Testes no Ecosystema Spring Boot
  - MockMVC
6. Testes de Integração
7. Testes de Aceitação (End-to-End)
  - Selenium
  - Cypress (Opcional)
8. Métricas para aferir cobertura de testes





# Metodologia: Como vamos aprender?

1. Live Coding: Programação ao vivo em sala.
2. Mão na Massa: Atividades práticas com Junit, Mockito e Selenium.
3. Coding Dojo: Prática em grupo para aprender TDD.
4. Projeto Real: Testando uma API Spring Boot completa.

# Avaliações



Os seguintes instrumentos serão utilizados para avaliação dos alunos:

1º Atividade avaliativa (**Introdução a Testes**): 05 pontos (1 pontos de avaliação atitudinal/participação)

2º Atividade avaliativa (**Junit básico**): 25 pontos (3 pontos de avaliação atitudinal/participação) (2 atividades)

3º Atividade avaliativa (**Mockito**): 15 pontos (2 pontos de avaliação atitudinal/participação)

# Avaliações



Os seguintes instrumentos serão utilizados para avaliação dos alunos:

3º Atividade avaliativa (**Junit Spring boot**): 20 pontos (3 pontos de avaliação atitudinal/participação)

5º Atividade avaliativa (**Testes de integração**): 10 pontos (2 pontos de avaliação atitudinal/participação)

6º Atividade avaliativa (**Testes de aceitação**): 10 pontos (2 pontos de avaliação atitudinal/participação)

7º Atividade avaliativa (**Métrica de testes**): 15 pontos (3 pontos de avaliação atitudinal/participação)

Workshop de Métricas

# Bibliografias



## **Bibliografia Básica:**

DELAMARO, M, E.; JINO, M.; MALDONADO, J. C. Introdução ao teste de software. São Paulo: Editora Campus, 2007.

FILHO, W. P. P. Engenharia de software: fundamentos, métodos e padrões. Rio de Janeiro: LTC, 2009.

KOSCIANSKI, A. Qualidade de software. São Paulo: Novatec, 2007.

## **Bibliografia Complementar:**

PILONE, D., MILES, R. Use a cabeça! Desenvolvimento de software. Rio de Janeiro: Alta Books, 2008.

SOMMERVILLE, I. Engenharia de software. São Paulo: Addison Wesley, 2007.

# Mãos a obra.....

