



**Universidade Federal do Ceará**  
**Centro de Ciências/Departamento de Computação**  
**Código da Disciplina:** CK0236  
**Professor:** Ismayle de Sousa Santos

**Aula 01**

# **Técnica de Programação II**

## **Apresentação da Disciplina**

---



**qpg4p5x**



**ismaylesantos@great.ufc.br**



**@IsmayleSantos**

# Agenda

- Apresentações
  - Objetivo da Disciplina
  - Ementa
  - Conteúdo
  - Bibliografia
  - Metodologia de Ensino
  - Avaliação
  - Background e Expectativas
-

# Apresentações

- Professor
  - <https://sites.google.com/site/ismaylesantos/>
- Turma

*Sejam  
bem  
Vindos*

---

# Apresentações

- **CKP0236 - Técnicas de Programação II**
    - Opcional
    - Pré-Requisito
      - CK0235 - Técnicas de Programação I
    - Carga Horária
      - 4 créditos - 64 h
    - Horários
      - SEG e QUA 16:00 às 18:00
    - Google Classroom
      - qpg4p5x
    - GitHub
      - <https://github.com/ismaylesantos/CK0236>
-

# Objetivo da Disciplina

- Apresentar aos alunos técnicas avançadas e boas práticas de programação e de desenvolvimento de software utilizadas na indústria de software.
    - Estimular o senso crítico do aluno sobre a escolha da técnica mais adequada para melhorar os indicadores de qualidade do software em desenvolvimento.
    - Fazer com que o aluno exercite os conteúdos apresentados por meio de práticas em laboratório simulando um ambiente real de desenvolvimento de software.
-

# Ementa

- Ambiente Integrado de Desenvolvimento (IDE). Versionamento de código. Teste unitário. Desenvolvimento guiado por teste. Testes automatizados. Depuração de software. Técnicas programação defensiva. Programação limpa (clean code). Princípios e técnicas de modularização e separação de interesses. Software code smells. Refatoração de software. Integração contínua.
-

# Conteúdo

## PARTE I:

- **Ambiente Integrado de Desenvolvimento (IDE)**
    - Eclipse, Visual Studio etc..
  - **Versionamento de Código**
    - Git, ChangeLog, Repositórios no Eclipse
  - **Desenvolvimento guiado por testes**
    - JUnit, TestNG
    - Reports
    - TDD versus ATDD
  - **Depuração de software**
  - **Testes automatizados**
    - Framework de Testes
    - Aplicações Móveis
    - Aplicações Web
-

# Conteúdo

## PARTE II:

- Técnicas programação defensiva
  - Programação limpa (clean code)
  - Princípios e técnicas de modularização e separação de interesses
-



# Conteúdo

## PARTE III:

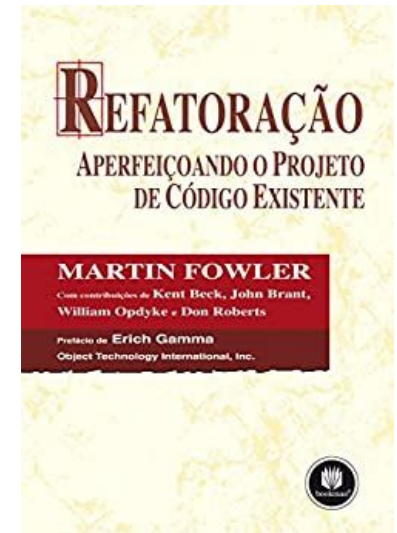
- **Software code smells**
    - **Test Smells**
  - **Refatoração de Software**
  - **Integração Contínua**
    - **Devops**
    - **Pipelines**
    - **CI/CD**
-

# Bibliografia



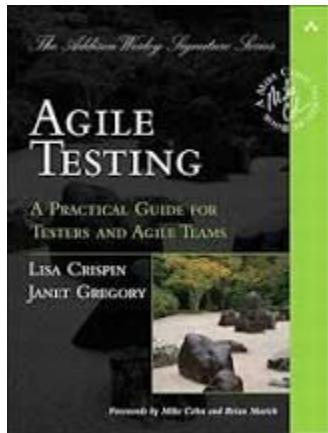
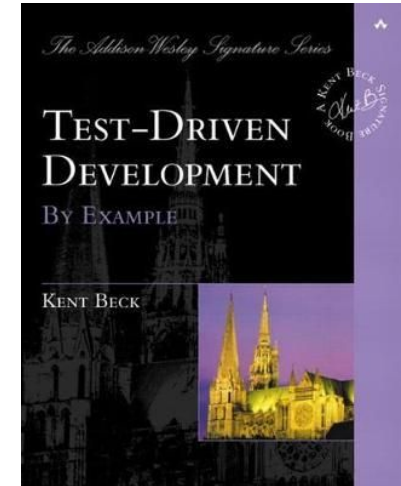
**MCCONNELL, Steve. Code Complete: um guia prático para a construção de software . 2. ed. Porto Alegre, RS: Bookman, 2005. ISBN 0735619670**

**FOWLER, Martin,. Refatoração: Aperfeiçoando o Projeto de Código Existente. Porto Alegre, RS: Bookman, 2004. ISBN: 8536303956.**



# Bibliografia

BECK, Kent,. **Test-Driven Development**. Pearson Education (US), 2003.



CRISPIN, Lisa; GREGORY, Janet. **Agile Testing: A Practical Guide for Testers and Agile Teams**. 1 ed., Addison-Wesley Professional , 2008.

# Metodologia de Ensino

- Aula expositiva dialogada
  - Aulas práticas
  - Trabalhos
    - Individuais
    - Em grupo
-

# Avaliação

A média semestral será calculada da seguinte forma:

- Atividades Práticas (40%)
- Projeto Final
  - Clean Code, Separação de Interesses, boas práticas (30%)
  - Testes automatizados (30%)

\*Alunos que não ficarem com média 7 ou superior farão avaliação final no fim do semestre

---

# Background e Expectativas

- Questionário para conhecê-los melhor :)

<https://forms.gle/YELeP4xBqQHEWeHX7>



# *Obrigado!*

*Por hoje é só pessoal...*

## **Dúvidas?**



qpg4p5x



ismaylesantos@great.ufc.br



@IsmayleSantos

---