

Plano da Disciplina: Engenharia de Software

1. Desenho da Disciplina

- **Nome:** Engenharia de Software
 - **Curso:** Licenciatura em Engenharia Informática, Ciência da Computação ou áreas correlatas
 - **Ano/Semestre:** 5º Ano / 1º Semestre
 - **Modalidade:** Presencial
 - **Carga Horária Semanal:** 5 horas
 - **Carga Horária Total:** 80 horas
-

2. Regime

- **Regime:** Semestral
 - **Frequência mínima:** 75% de presença nas aulas teórico-práticas
 - **Avaliação contínua** com provas, exercícios, desenvolvimento de projetos de software e implementação de metodologias de engenharia de software
-

3. Posição no Curso

- **5º Ano, 1º Semestre**
 - Disciplina fundamental para **modelagem, desenvolvimento e manutenção de software em larga escala**
-

4. Horário Semanal

- **2 horas teóricas** (fundamentos e metodologias de engenharia de software)
 - **1 hora teórico-prática** (análise e modelagem UML)
 - **2 horas práticas (laboratórios)** (implementação de sistemas e ferramentas de desenvolvimento)
-

5. Carga Horária Total

- **Total:** 80 horas
 - **Horas Teóricas:** 32h
 - **Horas Teórico-Práticas:** 16h
 - **Horas Práticas (Laboratórios):** 32h

6. Precedências Obrigatórias

- **Programação com Objetos**
-

7. Objetivos Gerais

- 1 Compreender **os princípios da Engenharia de Software e suas áreas de atuação.**
 - 2 Aplicar **metodologias de desenvolvimento de software para gerenciamento de projetos.**
 - 3 Desenvolver **habilidades de levantamento e gestão de requisitos.**
 - 4 Implementar **modelos UML para modelagem de sistemas.**
 - 5 Explorar **novos paradigmas como Desenvolvimento Ágil, DevOps e Model-Driven Architecture (MDA).**
-

8. Competências a Desenvolver

- ✓ Dominar **metodologias de desenvolvimento de software.**
 - ✓ Especificar e gerir **requisitos de software e documentação técnica.**
 - ✓ Utilizar **UML para modelagem de processos e sistemas.**
 - ✓ Desenvolver e validar **sistemas de software aplicando engenharia de software.**
 - ✓ Aplicar conceitos de **qualidade e manutenção de software.**
-

9. Conteúdo Temático e Bibliografia

Unidade 1: Introdução à Engenharia de Software (10h)

Pressman – Engenharia de Software, McGraw-Hill (1998)

- Definição e evolução da Engenharia de Software
- Áreas de intervenção da Engenharia de Software
- Conceitos básicos e ciclo de vida do software

Laboratório: Introdução a ferramentas CASE para engenharia de software

Unidade 2: Levantamento e Gestão de Requisitos (10h)

Software Engineering: A Practitioner's Approach – Roger Pressman (2000)

- Definição de requisitos
- Técnicas de levantamento e elicitação de requisitos
- Especificação e gestão de requisitos

Laboratório: Exercícios práticos de elicitação e documentação de requisitos

Unidade 3: Linguagem de Modelagem UML (10h)

UML, Metodologias e Ferramentas CASE – Alberto Silva, Carlos Videira (2001)

- Introdução à UML e evolução histórica
- Modelagem de Casos de Uso e Modelagem Estrutural
- Modelagem de comportamento e arquitetura

Laboratório: Modelagem UML de um sistema real utilizando ferramentas CASE

Unidade 4: Metodologias de Desenvolvimento de Software (10h)

Engenharia de Software – Sommerville (2011)

- Modelos tradicionais e ágeis
- Comparação entre desenvolvimento em cascata, espiral e incremental
- Implementação de metodologias ágeis

Laboratório: Simulação de desenvolvimento de software utilizando SCRUM

Unidade 5: Processo Unificado RUP (10h)

The Rational Unified Process – Philippe Kruchten (2004)

- Introdução ao Rational Unified Process (RUP)
- Estrutura e características principais
- Fases, iterações e workflows

Laboratório: Aplicação do RUP em um projeto realista

Unidade 6: Arquitetura e Qualidade de Software (15h)

Design Patterns – Erich Gamma et al. (1995)

- Padrões de análise e projeto
- Arquitetura orientada a serviços (SOA)
- Manutenção e refatoração de software

Laboratório: Implementação de padrões de projeto

Unidade 7: Desenvolvimento Ágil e DevOps (10h)

Scrum: A Arte de Fazer o Bom Trabalho na Metade do Tempo – Jeff Sutherland (2014)

- Princípios do Desenvolvimento Ágil
- Integração e entrega contínua (CI/CD)
- Automação de testes e deployment

Laboratório: Implementação de CI/CD em um projeto prático

Unidade 8: Projeto Final Integrado (10h)

Baseado nas competências adquiridas ao longo do semestre

- Desenvolvimento de um **projeto prático de engenharia de software utilizando metodologias aprendidas**
- Aplicação dos conceitos estudados na criação de um software funcional

Laboratório: Implementação e defesa do projeto final

10. Metodologia de Trabalho

Aulas teóricas para introdução aos conceitos fundamentais da engenharia de software

Exercícios práticos para reforço da modelagem e especificação de software

Laboratórios computacionais para desenvolvimento de sistemas reais

Projetos individuais e em grupo para aplicação prática dos conceitos estudados

11. Critério de Avaliação

- ✓ **40% - Trabalhos práticos e implementação de sistemas**
 - ✓ **30% - Projeto Final Integrado**
 - ✓ **20% - Participação e atividades complementares**
 - ✓ **10% - Prova escrita/técnica**
-

12. Bibliografia Básica

Roger Pressman – *Engenharia de Software*, McGraw-Hill (1998)

Roger Pressman, Darrel Ince – *Software Engineering: A Practitioner's Approach*, McGraw-Hill (2000)

Alberto Silva, Carlos Videira – *UML, Metodologias e Ferramentas CASE*, Centro Atlântico (2001)

Erich Gamma et al. – *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley (1995)

Jeff Sutherland – *Scrum: A Arte de Fazer o Dobro do Trabalho na Metade do Tempo*, Sextante (2014)