

Plano da Disciplina: Programação com Objetos

1. Desenho da Disciplina

- **Nome:** Programação com Objetos
 - **Curso:** Licenciatura em Engenharia de Software, Ciência da Computação ou áreas correlatas
 - **Ano/Semestre:** 3º 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, projetos práticos e análise de padrões de software
-

3. Posição no Curso

- **3º Ano, 1º Semestre**
 - Disciplina fundamental para **desenvolvimento de software orientado a objetos e engenharia de software**
-

4. Horário Semanal

- **2 horas teóricas** (fundamentos da Programação Orientada a Objetos)
 - **1 hora teórico-prática** (exemplos aplicados e modelagem de sistemas)
 - **2 horas práticas (laboratórios)** (implementação e experimentação em linguagens orientadas a objetos)
-

5. Carga Horária Total

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

6. Precedências Obrigatórias

- **Fundamentos de Programação**
-

7. Objetivos Gerais

- 1 Apresentar os **princípios da Programação Orientada a Objetos (POO)** e suas aplicações.
 - 2 Explorar conceitos como **encapsulamento, herança, polimorfismo e abstração**.
 - 3 Capacitar o aluno no **uso de linguagens POO, como Java e C++**.
 - 4 Introduzir **padrões de projeto (Design Patterns)** para desenvolvimento **eficiente de software**.
 - 5 Desenvolver habilidades práticas em **modelagem UML, refatoração e boas práticas de código**.
-

8. Competências a Desenvolver

- ✓ Compreender e implementar **conceitos fundamentais da POO**.
 - ✓ Projetar e desenvolver **sistemas utilizando linguagens orientadas a objetos**.
 - ✓ Aplicar **padrões de projeto (Design Patterns)** na construção de software **escalável**.
 - ✓ Modelar **sistemas utilizando diagramas UML**.
 - ✓ Integrar **testes automatizados no desenvolvimento de software**.
-

9. Conteúdo Temático e Bibliografia

Unidade 1: Fundamentos da Programação Orientada a Objetos (10h)

Head First Java – Kathy Sierra & Bert Bates

- Definição e motivação da POO
- Classes, objetos e métodos
- Encapsulamento e boas práticas

Laboratório: Implementação de classes e objetos em Java

Unidade 2: Estruturas Fundamentais em Java e C++ (10h)

Introdução à Programação Orientada a Objetos com C++ e Java – Ednaldo Pizzolato

- Operadores, expressões e controle de fluxo
- Manipulação de objetos
- Conversões de tipos e tratamento de erros

Laboratório: Construção de pequenas aplicações em Java e C++

Unidade 3: Construção e Organização de Código (10h)

Java 6 e Programação Orientada a Objetos – Fernando Martins

- Construtores, destrutores e métodos sobrecarregados
- Organização de código: bibliotecas e pacotes
- Diferença entre interface e implementação

Laboratório: Aplicação de modularidade no código

Unidade 4: Herança, Composição e Polimorfismo (15h)

Objetos, Abstração, Estruturas de Dados e Projeto usando C++ – Koffmann

- Desenvolvimento incremental de classes
- Utilização do modificador "final" e carregamento de classes
- Classes abstratas e interfaces

Laboratório: Implementação de polimorfismo e herança em Java

Unidade 5: Padrões de Projeto (Design Patterns) (15h)

Design Patterns – Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides

- Introdução aos padrões de design
- Singleton, Factory, Observer, Adapter
- Aplicação prática em projetos

Laboratório: Implementação de padrões de projeto

Unidade 6: Modelagem UML e Arquitetura de Software (10h)

Introdução à UML

- Diagramas de classes e sequência
- Modelagem de sistemas reais
- Refatoração e boas práticas de design

Laboratório: Modelagem de um sistema real utilizando UML

Unidade 7: Testes e Qualidade de Código (10h)

Head First Design Patterns – Eric Freeman

- Testes unitários e Test-Driven Development (TDD)

- Análise de código e refatoração
- Automação de testes

Laboratório: Implementação de testes automatizados

Unidade 8: Projeto Final Integrado (10h)

Baseado nas competências adquiridas ao longo do semestre

- Desenvolvimento de um **projeto prático de software utilizando POO e padrões de projeto**
- Modelagem UML, implementação e testes

Laboratório: Desenvolvimento e defesa do projeto final

10. Metodologia de Trabalho

Aulas teóricas para introdução aos conceitos de programação orientada a objetos

Exercícios práticos para reforço dos conceitos de modelagem e implementação

Laboratórios computacionais para experimentação e desenvolvimento de código

Projetos individuais e em grupo para aprofundamento dos conceitos

11. Critério de Avaliação

- ✓ **40% - Trabalhos práticos e exercícios laboratoriais**
 - ✓ **30% - Projeto Final Integrado**
 - ✓ **20% - Participação e atividades complementares**
 - ✓ **10% - Prova escrita e técnica**
-

12. Bibliografia Básica

Kathy Sierra, Bert Bates - *Head First Java*, O'Reilly (2005)

Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides - *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley (1995)

Ednaldo Brigante Pizzolato - *Introdução à Programação Orientada a Objetos com C++ e Java*, EdUfscar (2010)

Fernando Mário Junqueira Martins - *Java 6 e Programação Orientada a Objetos*, FCA (2009)

Koffmann, Elliot B. - *Objetos, Abstração, Estruturas de Dados e Projeto usando C++*, LTC