

PLANO DA DISCIPLINA - PROGRAMAÇÃO II

1. IDENTIFICAÇÃO DA DISCIPLINA

- **Nome:** Programação II
- **Curso:** Informática e Sistemas de Informação
- **Regime:** Semestral
- **Ano/Semestre:** 3º ano, 2º semestre
- **Carga Horária Total:** 80 horas
- **Horas Semanais:** 5 horas (2 teóricas, 2 teórico-práticas, 1 prática)
- **Precedência Obrigatória:** Programação I

2. OBJETIVOS GERAIS

- Aperfeiçoar os conhecimentos sobre programação orientada a objetos.
- Capacitar os alunos na criação e utilização de componentes em aplicações com interfaces gráficas.
- Introduzir o uso de sistemas de controle de versões.
- Aplicar padrões de design (Design Patterns) no desenvolvimento de software.

3. COMPETÊNCIAS A DESENVOLVER

- Implementação de conceitos avançados de programação orientada a objetos.
- Criação e manipulação de interfaces gráficas.
- Utilização de boas práticas no desenvolvimento de software.
- Uso de sistemas de controle de versões.
- Aplicação de padrões de design na construção de software escalável.

4. CONTEÚDO PROGRAMÁTICO

UNIDADE 1: Programação Orientada a Objetos Avançada (15h)

- Revisão dos conceitos básicos.
- Herança, polimorfismo e classes abstratas.
- Interfaces e métodos sobrecarregados.

UNIDADE 2: Desenvolvimento de Interfaces Gráficas (20h)

- Ferramentas e bibliotecas para interfaces gráficas.
- Sistema de coordenadas, cores e fontes.
- Comunicação entre componentes.

UNIDADE 3: Componentes Básicos e Avançados (20h)

- Elementos de UI: botões, menus, caixas de diálogo.
- Arrastar e largar (Drag-and-Drop) e validação de entradas.
- Visualização de listas e árvores.

UNIDADE 4: Entrada e Saída de Dados (15h)

- Sistema de arquivos e serialização.
- Ligação a bases de dados e manipulação de dados.
- Gerenciamento de persistência.

UNIDADE 5: Padrões de Design e Controle de Versão (10h)

- Introdução a padrões de design (Design Patterns).
- Controle de versão e colaboração com Git.
- Desenvolvimento de projetos aplicando boas práticas.

5. METODOLOGIA DE ENSINO

- Aulas teóricas com exemplos práticos.
- Desenvolvimento de projetos individuais e em grupo.
- Utilização de ferramentas modernas de desenvolvimento.
- Discussões sobre boas práticas e tendências na programação.

6. CRITÉRIOS DE AVALIAÇÃO

- **Trabalhos práticos e projetos:** 30%
- **Testes intermediários:** 30%
- **Exame final:** 40%
- **Frequência mínima:** 75% nas aulas práticas.

7. BIBLIOGRAFIA

Bibliografia Básica

- Horstmann, C. S., & Cornell, G. – *Core Java, Volume I - Fundamentos* (11ª ed.), Prentice Hall.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. – *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley.
- Eckel, B. – *Thinking in Java* (4ª ed.), Prentice Hall.

Bibliografia Complementar

- Bloch, J. – *Effective Java* (3ª ed.), Prentice Hall.
- Fowler, M. – *UML Distilled: A Brief Guide to the Standard Object Modeling Language* (3ª ed.), Addison-Wesley.
- Freeman, E., & Robson, E. – *Head First Design Patterns* (2ª ed.), O'Reilly Media.

- Bernd Bruegge & Allen Dutoit – *Object-Oriented Software Engineering Using UML, Patterns, and Java* (3ª ed.), Prentice Hall.