

DISCIPLINA: PROGRAMAÇÃO ORIENTADA A OBJETO	
<b>Código:</b>	
<b>Carga Horária Total: 80</b>	<b>CH Teórica: 40h   CH Prática: 40h</b>
<b>Número de Créditos:</b>	4
<b>Pré-Requisito:</b>	LÓGICA E LINGUAGEM DE PROGRAMAÇÃO
<b>Semestre:</b>	2
<b>Nível:</b>	Tecnologia
EMENTA	
Fundamentos do paradigma orientado a objetos (abstração, encapsulamento, classes, objetos, atributos, métodos e construtores), conceitos avançados de orientação a objetos (herança, polimorfismo, classes abstratas, interfaces e pacotes), metodologia de desenvolvimento orientada a objetos e aplicação dos conceitos de orientação a objetos através de uma linguagem de programação apropriada, incluindo Interface Gráfica com o Usuário.	
OBJETIVO	
Conhecer a metodologia de desenvolvimento orientada a objetos. Desenvolver aplicações de computadores através de uma linguagem de programação baseada no paradigma de programação orientada a objetos.	
PROGRAMA	
Unidade 1: A necessidade de planejamento para o desenvolvimento de software (10 ha): 1.1. O que é projeto de software e qual sua importância. 1.2. O processo de construção de um projeto. 1.3. Conceitos básicos sobre projetos de software: Robustez, Coesão, Facilidade de Uso. 1.4 Abstração, Complexidade, Hierarquia e Decomposição. Unidade 2: Conceitos de Orientação a objetos. (12 ha): 2.1. Programas Procedimentais x Programas Orientados a Objetos. 2.2. Objetos e Classes. 2.3. Herança e Polimorfismo. 2.4. Encapsulação. 2.5. Agregação e Composição. 2.6. Interfaces. Unidade 3: A Linguagem Java. (27 ha): 3.1. Mecanismos da Linguagem Java. 3.2. Identificadores, Palavras Reservadas e Tipos Primitivos. 3.3. Operadores, Expressões, Comandos e Controle de Fluxo. 3.4. Objetos e Classes. 3.5. Construtores. 3.6. Modificadores de Acesso e Armazenamento. 3.7. Exceções. Unidade 4: Estrutura de Dados. (25 ha): 4.1. Arrays. 4.2. Classes Auto-referenciadas. 4.3. Alocação Dinâmica de Memória. 4.4. Listas encadeadas. 4.5. Pilhas. 4.6. Filas. Unidade 5: Interface Gráfica com o Usuário.(21 ha): 5.1. Eventos e Interfaces. 5.2. Gerenciadores de Layout. 5.3. Componentes AWT / Swing. 5.4. Aplicação Prática dos Conceitos de Agregação, Composição, Generalização, Especialização, Polimorfismo. Unidade 6: Aplicações em Redes de Computadores. (25 ha): 6.1. Java I/O (Streams, Files, URL). 6.2. Sockets. 6.3. Threads.	
METODOLOGIA DE ENSINO	
A disciplina é desenvolvida no formato presencial: exposição teórica, práticas de laboratório, seminários e atividades a serem desenvolvidas extra-sala de aula. Os conteúdos das aulas serão detalhados conforme o cronograma do semestre.	
RECURSOS	
Laboratório de Microcomputação equipado com: <ul style="list-style-type: none"> <li>• Quadro branco;</li> <li>• Pincéis;</li> <li>• Projetor;</li> <li>• Computadores com aplicativos para desenvolvimentos de programas.</li> </ul>	
AVALIAÇÃO	
A avaliação é realizada de forma processual e cumulativa. A saber: avaliações escritas, trabalhos extra-sala de aula e dinâmicas em sala. A frequência é obrigatória, respeitando os limites de ausência previstos em	

lei. A sistemática de avaliação no ensino ocorrerá por todo o semestre letivo.. O semestre é dividido em duas etapas. Para cada uma delas, pelo menos uma avaliação deverá ser feita. A nota preliminar do semestre será calculada através da média ponderada das notas das duas etapas, sendo que a primeira etapa terá peso 2 e a segunda etapa terá peso 3. Se a nota preliminar do semestre for igual ou superior a 7, o aluno é considerado aprovado. Se o aluno obtiver nota maior ou igual a 3 e menor do que 7, ele terá direito a fazer uma avaliação final. A nota final do aluno será a média aritmética entre a nota preliminar do semestre e a nota da avaliação final. Se a nota final for maior ou igual a 5, o aluno é considerado aprovado.

#### **BIBLIOGRAFIA BÁSICA**

BARNES, David J.; KÖLLING, Michael. **Programação orientada a objetos com Java** : uma introdução prática usando o Blue J. São Paulo (SP): Pearson Prentice Hall, 2006. 368 p.

DEITEL, H. M.; DEITEL, P. J. **Java, como programar**. 3.ed. Porto Alegre (RS): Bookman, 2001. 1201 p.

HORSTMANN, Cay S.; CORNELL, Gary. **Core Java 2 - v.1**. São Paulo (SP): Makron Books/ Pearson Education, 2001. v.1.

PREISS, Bruno R. **Estruturas de dados e algoritmos: padrões de projetos orientados a objetos com Java**. Rio de Janeiro, RJ: Campus, 2000. 566 p. ISBN 85-7110-0693-0.

#### **BIBLIOGRAFIA COMPLEMENTAR**

FOWLER, Martin. **Refatoração: aperfeiçoando o projeto do código existente**. Porto Alegre (RS): Bookman, 2004. 365 p.

DEITEL, H. M., **C++: como programar**, Porto Alegre, RS : Bookman, 2004.

LEWAY, Laura, **Aprenda em 21 dias Java 2**, Rio de Janeiro, RJ : Campus, 1999.

SHLAER, Sally; MELLOR, Stephen J. **Análise de sistemas orientada para objetos**. São Paulo (SP): Makron Books do Brasil, 1990. 178 p.

SUMMERFIELD, Mark. **Programação em Python 3**: uma introdução completa à linguagem Python. Rio de Janeiro, RJ: Alta Books, 2015. 506 p. (Biblioteca do Programador). ISBN 9788576083849.

**Coordenador do Curso**

**Setor Pedagógico**