

Algoritmos e Lógica da Programação
------------------------------------

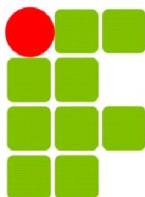
<b>Bibliografia</b>
---------------------

**BÁSICA**

Waldemar, C. Introdução a estruturas de dados. Elsevier, 2004  
 Guimarães, A. M. Algoritmos e estruturas de dados. LTC, 2008

**COMPLEMENTAR**

A. AHO, J. HOPCROFT E J. ULLMAN. Data Structures and Algorithms. Addison Wesley; 1st edition (January 11, 1983)  
 D. E. KNUTH. The Art of Computer Programming. AddisonWesley Professional; 3 edition (October 15, 1998)  
 CELES, W.; CERQUEIRA, R.; RANGEL, J. Introdução a Estruturas de Dados. 1ª edição. Rio de Janeiro: Campus, 2004.  
 M. T. GOODRICH, R. TAMASSIA. Estrutura de dados e algoritmos em Java. Quarta Edição. Editora Bookman, 2007.  
 T. H. CORMEN, C. E. LEISERSON, R. L. RIVEST E C. STEIN. Introdução a algoritmos. 3ª Edição. Editora Campus. 2012.



INSTITUTO FEDERAL DE  
 EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
 PARAÍBA

<b>Plano de Ensino</b>
------------------------

<b>Dados do Componente Curricular</b>
---------------------------------------

<b>Nome do Componente Curricular: Programação Orientada a Objetos</b>
---

<b>Curso: Técnico Subsequente em Informática</b>
--

<b>Série/Período: 2º semestre</b>
-----------------------------------

<b>Carga Horária: 6 a/s - 120 h/a - 100 h/r</b>	<b>Teóricas:</b>	<b>Práticas:</b>
---	------------------	------------------

<b>Docente Responsável: Elaine Cristina Juvino de Araújo</b>
--

### Ementa

Apresentar o paradigma de orientação a objetos como uma técnica para elaboração de projetos e implementação de sistemas de software de qualidade. Introduzir os recursos, características e suporte ferramental relacionados à linguagem de programação Java. Ao final do curso, o aluno deverá ser capaz de entender o conceito de abstração de dados, modelagem de sistemas utilizando-se objetos, herança, composição, polimorfismo e saber aplicar estes conceitos em situações práticas com uma linguagem de programação orientada a objeto.

### Objetivos

#### Geral

- Conhecer a metodologia de desenvolvimento orientada a objetos e uma linguagem de programação baseada em código aberto

#### Específicos

- Instalar o pacote de programas necessário para dar início à prática de programação;
- Configurar o ambiente de desenvolvimento para programação em Java;
- Importar bibliotecas para uso em projetos de programação;
- Explorar a documentação da API Java, criar classes e elaborar testes de unidade;
- Escrever programas utilizando dos recursos disponíveis para tratamento de erros e exceções;
- Conhecer fundamentos sobre o desenvolvimento de aplicações cliente/servidor;
- Conhecer uma linguagem de programação baseada em código aberto para desenvolvimento de aplicações voltadas para servidores;
- Conhecer uma linguagem de programação baseada em código aberto, voltada para realizar a interface entre o usuário e aplicação servidora;
- Apresentar a metodologia de desenvolvimento orientada a objetos, mostrando as técnicas e ferramentas para criação de programas usando linguagens de programação orientadas a objetos;
- Entender a principal diferença entre programas desenvolvidos utilizando a tradicional metodologia de programação estruturada e orientada a objeto;
- Familiarizar-se com os principais conceitos que determinam o entendimento do paradigma orientado a objeto;
- Valorizar a importância da utilização de boas práticas de programação na elaboração de código fonte.

### Conteúdo Programático

#### UNIDADE I

- Apresentação da disciplina e dos recursos disponíveis (2 h/a)
- Fundamentos da Linguagem Java (10 h/a)
  - Histórico e evolução da linguagem Java
  - Arquitetura da tecnologia Java

- Características da linguagem
  - Produtos e API's Java
  - Escrevendo, compilando e executando aplicações Java
  - Estado da arte em ambientes de desenvolvimento e execução
  - Abstração, objetos e visão geral de conceitos de POO
  - Utilização de suporte ferramental adequado e configuração do ambiente de trabalho
  - Automação de tarefas rotineiras com ANT e noções de *refactoring*
- Programação Orientada a Objetos com Java (20 h/a)
    - Classes e criação de objetos
    - Membros de classe: atributos e métodos (classe e instância)
    - Abstração de dados e encapsulamento
    - Construtores e suas características
    - Definindo mensagens e interface de objetos
    - Sobrecarga e sobreposição de métodos
    - Ciclo de vida dos objetos (instanciação à destruição)
    - Classes *Wrappers* (*Boolean, Character, Short, Integer, etc.*)
    - Estruturação e Manipulação de Objetos em Java
    - Herança e noções de Polimorfismo
    - Modelagem de Objetos usando a linguagem UML
  - Entrada e Saída Padrão de Dados em Java (4 h/a)
    - Entrada padrão de dados (classe Console)
    - Saída padrão de dados (System.out)
    - Entrada/Saída de dados GUI (classe JOptionPane)

## **UNIDADE II**

- Tipos, Literais, Operadores e Controle de Fluxo (8 h/a)
  - Palavras reservadas da linguagem
  - Constantes e variáveis
  - Tipos primitivos e de referência
  - Expressões
  - Coerção, conversão e promoção de tipos
  - Operadores: atribuição, aritméticos, relacionais, lógicos e bits
  - Estruturas de controle de fluxo
  - Operador '==' versus método equals(*Object o*).
  - Enumerações versus Variáveis de Classe;
- Encapsulamento e Visibilidade (8 h/a)
  - Definindo e refinando encapsulamento
  - Modificadores de visibilidade: *public, protected, default e private*
  - Criação de pacotes em Java
  - Importação de classes
- *Arrays* e *Strings* (4 h/a)
  - *Arrays* simples e multidimensionais
  - Ordenação de *arrays* (classe *Arrays*)
  - Características e manipulação de *Strings* e caracteres
  - Classes *String, StringBuilder* e *StringBuffer*
- Arquivos e Fluxos de Dados em Java (10 h/a)
  - Manipulação de dados em arquivos (pacote *java.io*)
  - Arquivos (classe *File*), fluxos de entrada e saída em Java
  - Leitura e gravação de Objetos e Textos em Java
- Tratamento de Erros e Exceções (12 h/a)

- Fundamentos acerca de tratamentos de erros e seus tipos
- Mecanismos *Try-Catch* e *Finally*
- Capturando e lançando exceções, finalizando exceções
- Exceções padrão em Java
- Criando novas exceções
- Exceções *Runnable*

### **UNIDADE III**

- Reutilização com Herança e Composição de Objetos (16 h/a)
  - Quando usar Herança ou Composição
  - Técnicas de composição e associação de objetos
  - Herança: vantagens e desvantagens sobre composição
  - Polimorfismo com herança e com composição
  - *Upcasting* e *Downcasting*.
  - Boas práticas de programação
  - Padrões de Projeto (essenciais) e boas práticas de programação
- Interfaces e Polimorfismo (10 h/a)
  - Fundamentos sobre polimorfismo
  - Aplicando polimorfismo com Interfaces
  - Classes abstratas e métodos abstratos
  - Mecanismo *Late binding* (vinculação dinâmica)
  - Interfaces e Herança múltipla em Java
- Conectividade e Aplicações em Rede com Java (6 h/a)
  - Classes Socket e ServerSocket
  - Objetos Distribuídos com RMI
- Coleções (10 h/a)
  - Coleções e API de estruturas de dados fundamentais
  - Tipos Genéricos
  - Listas, Mapas, Pilhas, Conjuntos e Filas
  - Métodos Genéricos

#### **Metodologia de Ensino**

- Aulas teóricas expositivas, aulas práticas, pesquisas individuais e em grupo, seminários, discussões e listas de exercícios. Além das atividades semanais em sala de aula, poderá haver até 20% da carga horária ministrada seguindo o modelo de ensino à distância usando soluções como, por exemplo, a plataforma "Moodle", com a realização de atividades, laboratórios e fóruns para dúvidas e discussões.

#### **Avaliação do Processo de Ensino e Aprendizagem**

- Provas escritas, trabalhos práticos e teóricos, seminários e listas de exercícios;

#### **Recursos Necessários**

- Quadro branco (negro) e pincel atômico (giz); Projetor multimídia; Softwares específicos para simulação de arquitetura de computadores.

#### **Pré-Requisitos**

- Algoritmos e Lógica da Programação

**Bibliografia****BÁSICA**

DEITEL, H.; DEITEL, P. **Java: Como Programar**. 8ª edição. Pearson Brasil, 2010.

SIERRA, K. **Use a cabeça! Java**. 2.ed. Alta Books, 2009.

**COMPLEMENTAR**

ECKEL, B. **Thinking in Java**. Prentice Hall, 2008. (<http://www.bruceeckel.com>).