



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO – UNIRIO.  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA – CCET  
ESCOLA DE INFORMÁTICA APLICADA – EIA

Programa de disciplina TP2 -Técnicas de Programação II		
CURSO: Bacharelado em Sistemas de Informação		
DEPARTAMENTO: Informática Aplicada		
DISCIPLINA: Técnicas de Programação II		
CÓDIGO: TIN0011 TIPO: Obrigatória		
CARGA HORÁRIA: 90 horas	Nº DE CRÉDITOS: 5 créditos	PERÍODO: 2º
PROFESSOR(ES): Paulo Sérgio Medeiros dos Santos (SIAPE 1064438)		
EMENTA: Tipos de dados compostos <del>Ponteiros</del> (Semântica de Referência) Implementação das estruturas de dados básicas (listas, filas e pilhas) Algoritmos recursivos		
PRÉ-REQUISITOS: Técnicas de Programação I – TIN0107		
OBJETIVOS DA DISCIPLINA: . Capacitar o aluno a desenvolver projetos utilizando conceitos básicos de modelagem, linguagens orientada a objeto e estruturas de dados.		
METODOLOGIA: <b>Exposição de conteúdo:</b> para apoiar o aluno nos tópicos da disciplina serão disponibilizados vídeos com aulas expositivas do conteúdo e com as soluções comentadas dos exercícios. <b>Aulas síncronas:</b> encontros síncronos para sanar dúvidas e reforçar tópicos apresentados nos vídeos e, principalmente, para resolução de exercícios de fixação. <b>Atividades assíncronas:</b> o aluno realizará atividades de programação em plataformas conhecidas (ex: <a href="https://www.urionlinejudge.com.br/">https://www.urionlinejudge.com.br/</a> , <a href="https://www.hackerrank.com/">https://www.hackerrank.com/</a> , <a href="https://www.techgig.com/">https://www.techgig.com/</a> , <a href="https://stepik.org/catalog?language=en&amp;q=java">https://stepik.org/catalog?language=en&amp;q=java</a> ). <b>Aprendizagem ativa:</b> ao longo de todo o curso os alunos resolverão, em conjunto, desafios de programação utilizando a dinâmica Dojo ( <a href="https://pt.wikipedia.org/wiki/Coding_Dojo">https://pt.wikipedia.org/wiki/Coding_Dojo</a> ).		
CONTEÚDO PROGRAMÁTICO: 1. Principais Características da Linguagem 2. Mecanismos de compilação e execução 3. Estrutura básica dos programas Java 4. Tipos Primitivos de Dados 5. Valores constantes 6. Declaração de Variáveis (sem e com inicialização de valores) 7. Declaração de constantes (final) 8. Operadores Aritméticos (+, -, *, /, %, ++, --) 9. Comandos de Saída 10. Comandos de entrada 11. Conversão de Tipos Numéricos 12. Operador de Conversão de Tipos (Casting) 13. Operadores de Atribuição Aritmética (+=, -=, *=, /=, %=)		

14. Operadores Relacionais (==, !=, >, >=, <, <=)
15. Operadores Lógicos (&&, ||, !)
16. Operador Condicional (?:)
17. Comandos de Controle de Fluxo
18. Recursão

#### Parte 2

1. Conceitos Básicos de Orientação a Objetos
2. Classes
3. Atributos
4. Operador new
5. Métodos Construtores
6. Princípio do Encapsulamento
7. Métodos de Acesso (get e set)
8. Métodos de Instância
9. Sobrecarga de métodos
10. Métodos e Atributos Estáticos
11. Classe Math
12. Classe String
13. Vetores

#### Parte 3

1. Relacionamentos entre Objetos
2. Relacionamentos 1:1
3. Relacionamentos 1:N
4. Coleções (ArrayList)
5. Herança
6. Polimorfismo
7. Classes e Métodos Abstratos
8. Interface
9. Tratamento de Exceções

#### Parte 4 (“bônus”)

1. Classe Object
2. Redefinição dos métodos equals(), toString() e clone()
3. Classes Wrapper
4. Generics
5. Ordenação de Listas
6. Arquivos

#### CRONOGRAMA:

Por conta da situação especial imposta pelo calendário suplementar, o cronograma do curso respeitará as necessidades da turma. A intenção será cobrirmos ao menos até a Parte 3. Da carga horária semanal de 6hs, o plano será ter 2hs de aulas síncronas. A depender da demanda da turma, em algumas semanas poderemos ter 4hs de aulas síncronas (respeitando o máximo de 50% de atividades síncronas no semestre).

SEMANA 1	Parte 1
SEMANA 2	Parte 1
SEMANA 3	Parte 1
SEMANA 4	Parte 2
SEMANA 5	Parte 2
SEMANA 6	Parte 2
SEMANA 7	Parte 2
SEMANA 8	Parte 2

SEMANA 9	Parte 3
SEMANA 10	Parte 3
SEMANA 11	Parte 3
SEMANA 12	Parte 3
SEMANA 13	Parte 4
SEMANA 14	Parte 4
SEMANA 15	Semana da divulgação de notas, execução e entrega de desafio de programação adicional que contará como avaliação final.

#### EXAMES E AVALIAÇÕES:

##### **Avaliação ao longo do período (ALP):**

Sessões assíncronas de dojos praticados em grupos de 3 pessoas realizados em horários combinados pelo próprio grupo devem ser gravados e enviados ao professor como parte da avaliação.

Desafios de programação, planejados como atividades assíncronas, também serão avaliativos.

Valor máximo: 10,0.

##### **Critério de aprovação:**

Se  $ALP \geq 5,0$ , a média é a ALP e o aluno está **APROVADO**.

Se  $ALP < 5,0$ , a média é a ALP e o aluno está **REPROVADO**.

#### FERRAMENTAS DIGITAIS UTILIZADAS:

Google Classroom

Google Meeting

Aulas gravadas

[replit.com](https://replit.com) – Compilador online para Java

#### BIBLIOGRAFIA ONLINE:

CAELUM. **Java e Orientação a Objetos**, disponível em

<https://www.caelum.com.br/apostila/apostila-java-orientacao-objetos.pdf>

DOWNEY, A. **Think Java: How to Think Like a Computer Scientist**, disponível em

<https://greenteapress.com/thinkajava/html/index.html>

#### BIBLIOGRAFIA BÁSICA:

ANSELMO, F. **Aplicando Lógica Orientada a Objetos em Java**, 2ª ed., Visual Books, 2005.

SANTOS, R. **Introdução à Programação Orientada a Objetos Usando Java**. 1ª ed., Campus, 2003.

#### BIBLIOGRAFIA COMPLEMENTAR:

DEITEL, H.; DEITEL, P. **Java - Como programar**, 10ª ed., Pearson, 2016.

Meyer, B. **Object-Oriented Software Construction**, 2ª ed., Prentice Hall, 1997.

Assinatura do professor: