



**UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO**  
**PRÓ-REITORIA DE ENSINO DE GRADUAÇÃO**

Rua Dom Manoel de Medeiros, s/n – Dois Irmãos 52171-900 Recife-PE  
Fone: 0xx-81-332060-40 proreitor@preg.ufrpe.br

**PLANO DE ENSINO**

**I – IDENTIFICAÇÃO**

CURSO: **Bacharelado em Sistemas de Informação**

MODALIDADE: **Presencial**

DISCIPLINA: **Introdução à Programação**

PRÉ-REQUISITO: --

( X ) OBRIGATÓRIA      ( ) OPTATIVA

DEPARTAMENTO: **Estatística e Informática**

PROFESSOR RESPONSÁVEL : **Cleviton Monteiro**

Ano: **2016**

Semestre Letivo:                      (X) Primeiro                      ( ) Segundo

Total de Créditos (se for o caso): **4**

Carga Horária: **90**

**II - EMENTA (Sinopse do Conteúdo)**

Fundamentos da construção de algoritmos e programas. Ambientes de programação: uso de uma linguagem de programação. Conceitos básicos: variáveis, operadores e expressões, estruturas de controle (atribuição, seleção, repetição). Dados estruturados: listas, cadeias, dicionários, tuplas). Subprogramas: funções, procedimentos. Parâmetros locais e globais. Recursão. Ordenação interna: bubblesort, inserção, shellsort, heapsort, quicksort. Pesquisa interna: seqüencial, binária. Modularização. Complexidade temporal de

algoritmos. Introdução a programação orientada a eventos. Introdução a programação orientada a objetos. Introdução a programação orientada a aspectos. Projeto: desenvolvimento de um programa de porte médio.

### **III - OBJETIVOS DA DISCIPLINA**

O objetivo principal desta oferta de disciplina é fornecer ao estudante a introdução à lógica e às técnicas e principais estruturas de programação.

### **IV - CONTEÚDO PROGRAMÁTICO**

#### 1. Algoritmo e linguagens

1.1 Algoritmo (fluxograma). 1.2 Instruções de um algoritmo. 1.3 Tipos de linguagens. (máquina, simbólica, abstrata). 1.4 Programa, compilação, interpretação e execução.

#### 2. Ambiente de programação

2.1 Conceitos básicos de uma linguagem: variáveis, tipos, estruturas de controle. 2.2 Declaração de tipo simples. 2.3 Comandos de entrada e saída. 2.4 Comando de atribuição. 2.5 Operadores e expressões. 2.6 Exercícios em laboratório.

#### 3. Tipos estruturados

3.1 Listas 3.2 Filas 3.3 Array unidimensional (vetor). 3.4 Array bidimensional (matriz). 3.5. Arquivos. 3.6. Exercícios em laboratório

#### 4. Subprogramas

4.1 Função com parâmetro. 4.2 Subroutine com parâmetro. 4.3 Passagem de parâmetro. 4.4. Exercício em laboratório

#### 5. Algoritmos

5.1 Pesquisa sequencial e binária 5.2 bubblesort 5.3 inserção 5.4 shellsort 5.5 heapsort 5.6 quicksort.

#### 6. Noções de Complexidade e Paradigmas de Programação

6.1 Notação  $o$ ,  $O$ ,  $\Omega$  e  $\Theta$ . 6.2 Introdução a programação orientada a eventos. 6.3 Introdução a programação orientada a objetos. 6.4 Introdução a programação orientada a aspectos.

#### 7. Projeto de Implementação.

## V – MÉTODOS DIDÁTICOS DE ENSINO

- ( X ) Aula Expositiva
- ( ) Seminário
- ( X ) Leitura Dirigida
- ( X ) Demonstração (prática realizada pelo Professor)
- ( X ) Laboratório (prática realizada pelo aluno)
- ( ) Trabalho de Campo
- ( X ) Execução de Pesquisa
- ( X ) Outra. Especificar: Estudo de Caso, Solução de Problemas, Mini testes.

## VI - CRITÉRIOS DE AVALIAÇÃO

FORMAS DE ACOMPANHAMENTO DO ALUNO DURANTE O SEMESTRE:

**Serão realizadas avaliações para compor as notas referentes a 1VA e 2VA:**

NOTA 1V.A.: Trabalhos escritos (50%) + trabalhos escolares práticos (50%)

NOTA 2V.A.: Trabalhos escritos (50%) + trabalhos escolares práticos (50%)

NOTA 3V.A.: Prova escrita com todo o conteúdo (100%)

NOTA FINAL: Prova escrita com todo o conteúdo (100%)

## CRONOGRAMA

Aula	CONTEÚDO
<b>1ª. Aula (2h)</b>	Apresentação e introdução
<b>2ª. Aula (2h)</b>	Algoritmos - Introdução
<b>3ª. Aula (2h)</b>	Algoritmos - Conceitos básicos
<b>4ª. Aula (2h)</b>	Algoritmos - Conceitos básicos
<b>5ª. Aula (2h)</b>	Algoritmos - Exercícios
<b>6ª. Aula (2h)</b>	Python - Introdução
<b>7ª. Aula (2h)</b>	Python - Introdução
<b>8ª. Aula (2h)</b>	Python - Condicionais e loops
<b>9ª. Aula (2h)</b>	Python - Condicionais e loops
<b>10ª. Aula (2h)</b>	Python - Condicionais e loops
<b>11ª. Aula (2h)</b>	Exercícios - Condicionais e loops
<b>12ª. Aula (2h)</b>	Mini teste 1 (valor: 1 ponto) - Algoritmos e Python (condicionais e loops)
<b>13ª. Aula (2h)</b>	Python - Strings

<b>14ª. Aula (2h)</b>	Python - Strings
<b>15ª. Aula (2h)</b>	Python - Listas
<b>16ª. Aula (2h)</b>	Python - Listas
<b>17ª. Aula (2h)</b>	Python - Listas
<b>18ª. Aula (2h)</b>	Exercícios - Strings e Listas
<b>19ª. Aula (2h)</b>	Mini teste 2 (valor: 1 ponto) - Strings e Listas
<b>20ª. Aula (2h)</b>	Busca/pesquisa sequencial e binária
<b>21ª. Aula (2h)</b>	Aula prática – Projeto 1
<b>22ª. Aula (2h)</b>	Prova 1 (VA) (valor: 7 pontos)
<b>23ª. Aula (2h)</b>	Python - Funções
<b>24ª. Aula (2h)</b>	Python - Funções
<b>25ª. Aula (2h)</b>	Apresentação do projeto 1 (valor: 1 ponto)
<b>26ª. Aula (2h)</b>	Python - Dicionários
<b>27ª. Aula (2h)</b>	Python - Dicionários
<b>28ª. Aula (2h)</b>	Exercícios - Funções e Dicionários
<b>29ª. Aula (2h)</b>	Mini teste 3 (valor: 1 ponto) - Funções e Dicionários
<b>30ª. Aula (2h)</b>	Python - Arquivos
<b>31ª. Aula (2h)</b>	Python - Arquivos
<b>32ª. Aula (2h)</b>	Aula prática – Projeto 2
<b>33ª. Aula (2h)</b>	Python - Ordenação
<b>34ª. Aula (2h)</b>	Python - Ordenação
<b>35ª. Aula (2h)</b>	Exercícios - Arquivos e ordenação
<b>36ª. Aula (2h)</b>	Mini-teste 4 (valor: 1 ponto) - Arquivos e ordenação
<b>37ª. Aula (2h)</b>	Python – OO
<b>38ª. Aula (2h)</b>	Python - OO
<b>39ª. Aula (2h)</b>	Python - OO
<b>40ª. Aula (2h)</b>	Exercícios - OO
<b>41ª. Aula (2h)</b>	Python - OO (Parte 2)
<b>42ª. Aula (2h)</b>	Python - OO (Parte 3)
<b>43ª. Aula (2h)</b>	Exercícios - Python OO (Partes 2 e 3)
<b>44ª. Aula (2h)</b>	Prova 2 (VA) (valor: 7 pontos)
<b>45ª. Aula (2h)</b>	Prova 3 (VA) (valor: 7 pontos)

## **VIII – BIBLIOGRAFIA**

### **BÁSICA:**

1. Menezes, Nilo. Introdução à Programação com Python: algoritmos e lógica de programação para iniciantes. Editora Novatec, 2010.
2. Lopes, Anita e Garcia, Guto. Introdução a Programação. Editora Campus, 2002.
3. Cormen, Thomas H. et. al. Algoritmos: Teoria e Prática. Editora Campus, 2002.
4. Van Rossum, Guido. Tutorial de Python. Disponível gratuitamente em <http://python.org/>, 2004.

### **COMPLEMENTAR:**

1. Ziviani, Nivio. Projeto de Algoritmos. Editora Nova Fronteira, 2004.
2. Sebesta, Robert W. Conceitos de Linguagens de Programação. Bookman, 2001.
3. Downey, Allen B. Python for Software Design: How to Think Like a Computer Scientist. Cambridge University Press, 2009.
4. Nieto, T. R. Internet & World Wide Web. Como Programar. Bookman, 2003.
5. Deitel, Harvey M. et. al. XML Como Programar. Bookman, 2003.

Recife, 07 de Março de 2016

**Cleviton Monteiro**  
*Professor Responsável*