



**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: **Laboratório de Programação**

PRÉ-REQUISITO: **Laboratório de Informática**

( X ) OBRIGATÓRIA      ( ) OPTATIVA

DEPARTAMENTO: **Estatística e Informática**

PROFESSOR RESPONSÁVEL : **Rodrigo Gabriel Ferreira Soares**

Ano: **2015.1**

Semestre Letivo:                      ( X ) Primeiro                      ( ) Segundo

Total de Créditos (se for o caso): **3**

Carga Horária: **60**

**II - EMENTA (Sinopse do Conteúdo)**

Ambientes de Programação. Projeto, Implementação e Teste de Software. Projeto de Sistemas de Software.

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

O objetivo principal desta oferta de disciplina é fornecer ao estudante oportunidade de trabalhar em atividades práticas e na resolução de problemas através da programação e do uso de estruturas de dados.

### **IV - CONTEÚDO PROGRAMÁTICO**

1. Ambientes de Programação
  - 1.1 Ferramentas
  - 1.2 Ambientes Integrados de Desenvolvimento
2. Projeto de Software
  - 2.1 Descrição em Linguagem de modelagem UML
  - 2.2 Documentação de Projeto de Software
3. Implementação de Software
  - 3.1 Codificação
  - 3.2 Práticas de Programação
  - 3.3 Práticas de Documentação
4. Teste de Software
  - 4.1 Técnicas de Teste de Software
  - 4.2 Ferramentas de teste de software
5. Prática de Projeto de Sistemas

### **V – MÉTODOS DIDÁTICOS DE ENSINO**

- Aula Expositiva
- Seminário
- Leitura Dirigida
- Demonstração (prática realizada pelo Professor)
- Laboratório (prática realizada pelo aluno)
- Trabalho de Campo
- Execução de Pesquisa
- Outra. Especificar: \_\_\_\_\_

## VI - CRITÉRIOS DE AVALIAÇÃO

FORMAS DE ACOMPANHAMENTO DO ALUNO DURANTE O SEMESTRE:

NOTA 1V.A. (Série de trabalhos práticos)

NOTA 2V.A. (Série de trabalhos práticos e projeto de desenvolvimento de software)

NOTA 3V.A. (Prova com todo o conteúdo - 100% da nota)

NOTA FINAL (Prova com todo o conteúdo - 100% da nota)

<b>CRONOGRAMA</b>	
DATA	CONTEÚDO
<b>1ª. Aula (2h)</b>	Apresentação, Ambientes de Desenvolvimento de Software
<b>2ª. Aula (2h)</b>	Ferramentas de desenvolvimento
<b>3ª. Aula (2h)</b>	Conceitos básicos da linguagem Python e orientação a objetos
<b>4ª. Aula (2h)</b>	Conceitos básicos da linguagem Python e orientação a objetos
<b>5ª. Aula (2h)</b>	Funções comuns e recorrência
<b>6ª. Aula (2h)</b>	Funções comuns e recorrência
<b>7ª. Aula (2h)</b>	Implementação de algoritmos de ordenamento
<b>8ª. Aula (2h)</b>	Implementação de algoritmos de ordenamento
<b>9ª. Aula (2h)</b>	Implementação de listas, pilhas e filas
<b>10ª. Aula (2h)</b>	Implementação de exercícios (1º teste da 1ª VA)
<b>11ª. Aula (2h)</b>	Implementação de tabelas hash
<b>12ª. Aula (2h)</b>	Implementação de tabelas hash
<b>13ª. Aula (2h)</b>	Implementação de árvores
<b>14ª. Aula (2h)</b>	Implementação de árvores
<b>15ª. Aula (2h)</b>	Implementação de árvores
<b>16ª. Aula (2h)</b>	Implementação de exercícios (2º teste da 1ª VA)
<b>17ª. Aula (2h)</b>	definição de projetos dos alunos
<b>18ª. Aula (2h)</b>	Introdução à Engenharia de requisitos
<b>19ª. Aula (2h)</b>	Ferramentas CASE
<b>20ª. Aula (2h)</b>	Introdução à UML
<b>21ª. Aula (2h)</b>	Introdução à UML
<b>22ª. Aula (2h)</b>	Introdução à Análise de requisitos
<b>23ª. Aula (2h)</b>	Introdução à Análise de requisitos
<b>24ª. Aula (2h)</b>	Introdução à Análise de casos de uso
<b>25ª. Aula (2h)</b>	Introdução à Análise de casos de uso
<b>26ª. Aula (2h)</b>	Acompanhamento de projetos

<b>27ª. Aula (2h)</b>	Acompanhamento de projetos
<b>28ª. Aula (2h)</b>	Apresentação de projetos (2ª VA)
<b>29ª. Aula (2h)</b>	Apresentação de projetos (2ª VA)
<b>30ª. Aula (2h)</b>	3ª VA

## **VIII – BIBLIOGRAFIA**

### **BÁSICA:**

1. Martin Fowler et al. UML Essencial. Editora Bookman. 2005.
2. Booch, Grady et.al. UML – Guia do Usuário. Campus. 2006.
3. Gross, Hans – Gerhard. Component-Based Software Testing with UML. Springer Verlag. 2004.
4. Cormen, Thomas H. et. al. Algoritmos: Teoria e Prática. Editora Campus, 2002.
5. Sebesta, Robert W. Conceitos de Linguagens de Programação. Bookman, 2001.
6. Van Rossum, Guido. Tutorial de Python. Disponível gratuitamente em <http://python.org/>, 2004.

### **COMPLEMENTAR:**

1. Cardoso, Caíque. UML na Prática. Ciência Moderna. 2003.
2. Ziviani, Nivio. Projeto de Algoritmos. Thomson Learning, 2004.

Recife, 12 de Fevereiro de 2015.



Rodrigo Gabriel Ferreira Soares  
*Professor Responsável*