|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Unidade Universitária:**  Faculdade de Computação e Informática | | | | | |
| **Curso:**  CIÊNCIA DA COMPUTAÇÃO | | | | | **Núcleo Temático:**  Algoritmos e Programação |
| **Disciplina:**  ALGORITMOS E PROGRAMAÇÃO I | | | | | **Código da Disciplina:**  ENEC50011 |
| **Professor(es):** JEAN MARCOS LAINE  FABIANA ARANTES MATHEUS | | | **DRT:** 115337-7 | | **Etapa:**  01 |
| **Carga horária:**  4h/a | | ( 02 ) Sala de Aula  ( 02 ) Laboratório | | | **Semestre Letivo:**  1o Semestre/2020 |
| **Ementa:**  Estudo e desenvolvimento de algoritmos envolvendo comandos de atribuição, condicionais e de repetição, tendo com ênfase a resolução de problemas em ordem crescente de complexidade. Implementação de algoritmos utilizando linguagem de programação imperativa. | | | | | |
| **Objetivos:** | | | | | |
| **Fatos e Conceitos** | **Procedimentos e Habilidades** | | | **Atitudes, Normas e Valores** | |
| - Conhecer o conceito de algoritmo computacional - Identificar passos para soluções de problemas elementares e formalizá-los através de algoritmos  - Avaliar e comparar soluções algorítmicas para problemas elementares  - Conhecer estruturas de programação de uma linguagem imperativa e aplicá-los na implementação de algoritmos | - Construir algoritmos computacionais para problemas elementares  - Implementar algoritmos em uma linguagem de programação imperativa   - Configurar e utilizar ambientes de implementação de algoritmos  - Simular implementações de algoritmos para avaliação de funcionamento e detecção de erros | | | - Reconhecer a importância dos algoritmos para resolução de problemas.  - Reconhecer a importância da linguagem de programação na implementação de algoritmos.  - Reconhecer a importância do teste de algoritmos.  - Reconhecer a área de programação como um suporte essencial na construção de sistemas computacionais.  - Perceber e superar dificuldades inerentes ao pensamento algorítmico. | |
| **Conteúdo Programático:**  0. Apresentação do Plano de Ensino  1. UNIDADE I: FUNDAMENTOS DE ALGORITMOS E PROGRAMAÇÃO  1.1. Problemas e soluções  1.2. Algoritmos e exemplos de notação (fluxogramas, pseudocódigo)  1.3. Programas e algoritmos  2. UNIDADE II: VARIÁVEIS, TIPOS DE DADOS, EXPRESSÕES, ATRIBUIÇÃO E ESTRUTURA SEQUENCIAL  2.1. Constantes e variáveis  2.2. Tipos (numéricos, booleanos, caracter)  2.3. Operadores e expressões matemáticas  2.4. Operador de atribuição  2.5. Estrutura sequencial  2.6. Problemas envolvendo variáveis, tipos de dados, expressões, atribuição e estrutura sequencial  3. UNIDADE III: ESTRUTURAS DE DECISÃO  3.1. Operadores relacionais e lógicos e suas tabelas  3.2. Estrutura de Seleção Simples  3.3. Estrutura de Seleção Composta  3.4. Encadeamento de estruturas de decisão  3.5. Problemas envolvendo estruturas de decisão  4. UNIDADE IV: ESTRUTURAS DE REPETIÇÃO  4.1. Estrutura de repetição com teste no início  4.2. Estrutura de repetição com teste no final  4.3. Estrutura de repetição com variável de controle  4.4. Encadeamento de estruturas de repetição  4.5. Problemas com estruturas de repetição  5. UNIDADE V: FUNÇÕES  5.1. Definição e Uso de Funções  5.2. Aplicações em problemas  6. UNIDADE VI: REPRESENTAÇÃO DE LISTAS  6.1. Notação de Listas  6.2. Operações sobre Listas  6.3. Exemplos e Aplicações | | | | | |
| **Metodologia:**   * Aulas expositivas * Aulas práticas em laboratórios * Utilização do ambiente Moodle * Listas de Exercícios e Projetos Práticos | | | | | |
| **Critério de Avaliação:**  **Nota 1** (N1) composta de: • Prova parcial 1 (**P1**) escrita, individual e sem consulta (70%) – NOTA A (TIA) • Atividades de Laboratório (**Lab1**) (30%) – NOTA B (TIA)  **Nota 2** (N2) composta de: • Prova parcial 2 (**P2**) escrita, individual e sem consulta (70%) – NOTA F (TIA)  • Atividades de Laboratório (**Lab2**) (30%) – NOTA G (TIA)   **MI = (N1 + N2)/2 + NP**  • Nota de participação (**NP**): até um ponto (0 a 1.0) – lista de exercícios no *URI Online Judge*.  Se MI ≥ 7.5, aluno está APROVADO e a Média Final (MF) = MI.  Caso contrário, poderá fazer uma PROVA SUBSTITUTIVA (SUB), que substituirá a menor nota  entre N1 e N2.  Ou, se o aluno ainda não for aprovado, poderá fazer a PROVA FINAL (PF):  MF = (MI + PF)/2  Se MF ≥ 6.0, aluno está APROVADO. Caso contrário, está REPROVADO. | | | | | |
| **Bibliografia Básica:**   |  | | --- | | DIERBACH, C. Introduction to Computer Science Using Python: A Computational Problem-Solving Focus.1.ed. New York: Wiley, 2012.  MENEZES, N.N.C. Introdução à Programação com Python: algoritmos e lógica de programação para iniciantes. São Paulo: Novatec, 2a.ed, 2014.  ZELLE, J.M. Python Programming: An Introduction to Computer Science, 2nd Edition, Franklin, Beedle & Associates Inc, 2009. | | | | | | |
| **Bibliografia Complementar:**  - FORBELLONE, A. L. V.; EBERSPACHER, H. F. Lógica de Programação: A Construção de Algoritmos e Estrutura de Dados. 3. ed. Sao Paulo: Pearson Prentice Hall, 2012.  - KINSLEY, H.; MCGUGAN, W. Introdução ao Desenvolvimento de Jogos em Python com PyGame. São Paulo: Novatec, 2015.  - LOPES, A.; GARCIA, G.; Introdução a Programação: 500 Algoritmos. Rio de Janeiro: Editora Campus, 2002.  - PAYNE, B. Ensine seus filhos a programar. São Paulo: Novatec, 1a. ed. 2015.  - PIVA Jr., D.; NAKAMITI, G.S., ENGELBRECHT, A.M. Algoritmos e Programação de Computadores. Rio de Janeiro: Editora Elsevier Ltda, 2012. | | | | | |