

Programação I - 2022/1

INF01106

Profa. Dra. Annabell D.R. Tamariz

Universidade Estadual do Norte Fluminense Darcy Ribeiro

16 de março de 2022

Programação I - 2022/1

INF01106

Profa. Dra. Annabell D.R. Tamariz

Universidade Estadual do Norte Fluminense Darcy Ribeiro

16 de março de 2022

Horários e Avaliações

Ementa

Introdução

Resolução de problemas

Hardware e Software

Hardware

Software

Horários - Avaliações

- ▶ Terça e Quinta-feira das 16:00 as 18:00 no Auditório do P5.
- ▶ **20 até 24/Junho/2022: CONFICT**
- ▶ Prova 1: **05/Maio/2022** - Quinta-Feira
- ▶ Prova 2: **23/Junho/2022** - Quinta-Feira
- ▶ Prova Final: **7/Julho/2022**

Média (M) $\geq 6,0$ pontos - Aprovado.

Se M $\geq 4,0$ pode fazer PF.

Nota final (NF) = $(M + PF) / 2$

Aprovação na disciplina $\Rightarrow NF \geq 5$

e Presença $\geq 75\%$.

Metodologia

- ▶ Complementação no Classroom.
- ▶ Aula invertida, para resolução de exercícios.
- ▶ Provas Presenciais: 2 + Final.
- ▶ Encontros para dúvidas pelo Meet.
- ▶ Contacto pelo e-mail *annabell_pos@pq.uenf.br* ou recados no classroom.

Objetivos da Disciplina

1. Desenvolver um raciocínio aplicado na formulação e resolução de problemas computacionais;
2. Ensinar como abordar e resolver problemas computacionais.

Sub-objetivos importantes:

- ▶ Aprender a escrever programas de computador.
- ▶ Aprender aspectos básicos de uma linguagem de programação, como ferramenta didática.
- ▶ O ato de programar é uma ferramenta útil para trabalhar o raciocínio lógico, bem como tornar mais concretos outros conceitos comuns em ciência da computação.

Objetivos da Disciplina

- ▶ O ensino de uma linguagem de programação é um bônus concreto de aprendizagem.
- ▶ Uma vez adquirida familiaridade com uma linguagem de programação, não é difícil utilizar outra. Muitas vezes, basta apenas ler o manual.
- ▶ Aprender um pouco da linguagem C, Python ou outras;
- ▶ Nas avaliações propostas será cobrada a aplicação dos conceitos vistos nas aulas teóricas.

IMPORTANTE!!!!

O verdadeiro aprendizado será a familiaridade com os conceitos e princípios envolvidos em programar.



Objetivos da Disciplina

- ▶ Nesta disciplina não vamos aprender linguagem nenhuma porque elas não são o foco. Embora alguns exemplos práticos serão apresentados em linguagem C, Python, por exemplo.
- ▶ Nosso objetivo é desenvolver um raciocínio aplicado na formulação e resolução de problemas computacionais.

Horários e Avaliações

Ementa

Introdução

Resolução de problemas

Hardware e Software

Hardware

Software

EMENTA ORIGINAL - SISTEMA ACADÊMICO

► Ementa:

1. Resolução de problemas e desenvolvimento de algoritmos;
2. Estruturação de programação;
3. Algoritmos;
4. Procedimentos, funções e passagem de parâmetros;
5. Conceitos de recursão e sua aplicação;
6. Linguagem estruturada;
7. Estratégias de depuração de programas.

Conteúdo Programático

1. Resolução de problemas e desenvolvimento de algoritmos
 - 1.1 Análise do problema, estratégias de solução, representação, e documentação.
 - 1.2 Algoritmos em diagramas de fluxo, pseudo-códigos
2. Estruturação de programas
 - 2.1 Tipos de dados simples e avançados: vetores, matrizes, cadeias de caracteres e registros, conjuntos, arquivos e estruturas dinâmicas(ponteiros), e suas aplicações.
3. Modularização de programas
 - 3.1 Procedimentos, funções e passagens de parâmetros
4. Conceitos de recursão e sua aplicação. Tipos de dados avançados.
5. Prática/Ensino de uma linguagem estruturada.
6. Depuração de programas. Laboratórios de programação

Bibliografia

- ▶ Ascencio, A.F.G.; Campos, E.A.V. - Fundamentos da Programação de Computadores - Algoritmos, Pascal e C/C++, Prentice Hall, 2003.
- ▶ Forbellone, A.L.V.; Eberspacher, H.F. - Lógica de Programação, 2ª Edição, Makron Books, 2000.
- ▶ DAN SWAIT JR., J. Fundamentos Computacionais - Algoritmos e Estruturas de Dados, Makron Books, McGraw-Hill, 1991.
- ▶ GUIMARÃES, A.M.; LAJES, N.A.C. Algoritmos e Estruturas de Dados, Livros Técnicos e Científicos, Editora S/A., 1985.
- ▶ TREMBLAY, J.P.; BUNT, R.B. Ciências dos Computadores: Uma Abordagem Algorítmica, São Paulo, McGraw-Hill, 1983.
- ▶ SCHNEIDER, WEINGART, PERLMAN An Introduction to Programming and Problem Solving with PASCAL, John Wiley & Sons, 1978.

Horários e Avaliações

Ementa

Introdução

Resolução de problemas

Hardware e Software

Hardware

Software

INTRODUÇÃO

Antes de iniciarmos a discussão sobre os conceitos de linguagens de programação, precisamos considerar aspectos preliminares.

- ▶ Algumas razões pelas quais os estudantes de ciência da computação e os desenvolvedores de software profissionais devem estudar conceitos gerais sobre o projeto e a avaliação de linguagens.
- ▶ Essa discussão é particularmente valiosa para quem acredita que um conhecimento funcional de uma ou duas linguagens de programação é suficiente para cientistas da computação.

Razões para estudar Conceitos de LP

- ▶ Maior capacidade para expressar ideias
- ▶ Embasamento aprimorado para escolher linguagens apropriadas
- ▶ Maior capacidade de aprender novas linguagens;
- ▶ Melhor compreensão do significado da implementação
- ▶ Melhor uso de linguagens já conhecidas
- ▶ Avanço geral da computação

Algoritmos no dia a dia

- ▶ Encontrar o caminho mais curto no Google mais => Algoritmos de caminho.
- ▶ Encontrar um livro em uma biblioteca => Algoritmos de busca.
- ▶ Organizar livros em uma prateleira => Algoritmos de ordenação.
- ▶ Redes sociais e Recomendações de vídeos no YouTube => Algoritmos da teoria dos Grafos.
- ▶ Embarcar em um ônibus => Algoritmos de fila (queue).
- ▶ Inverter uma strings => Algoritmos de pilha (stack).

Áreas de Aplicação / Domínios de Programação

► Aplicações Científicas:

1. Grande número de cálculos de ponto flutuante;
2. Uso de matrizes/*arrays*
3. Linguagem Fortran

► Aplicações de Negócios:

1. Produção de relatórios, uso de números decimais e caracteres
2. Linguagem COBOL

► Na Inteligencia Artificial:

1. Manipulação de Símbolos em vez de números;
2. Uso de listas encadeadas.
3. Linguagem LISP ou Prolog.

Áreas de Aplicação / Domínios de Programação

- ▶ Programação de Sistemas:
 1. Precisa de eficiência devido ao uso contínuo,
 2. Linguagem C
- ▶ Software para Web: Coleção eclética de linguagens, tais como
 1. Marcação (por exemplo, HTML),
 2. Scripts (por exemplo, PHP),
 3. Uso geral (por exemplo, Java)

Horários e Avaliações

Ementa

Introdução

Resolução de problemas

Hardware e Software

Hardware

Software

Resolução de Problemas

Vocês aprenderão como usar um computador para resolver problemas.

- ▶ Definiremos um problema a ser resolvido,
- ▶ Descreveremos uma solução imperativa e
- ▶ Implementaremos esta solução criando um programa.

Uma solução imperativa descreve uma sequencia de comandos e passos que devem ser executados para se resolver um problema.

- ▶ Uma solução imperativa para um determinado problema é conhecida como **algoritmo** para se resolver o problema.

Exemplo - Problema

Suponha o problema de se encontrar a raiz quadrada de x .

- ▶ Uma solução imperativa (algoritmo) para encontrar \sqrt{x} de forma aproximada:

1. Comece com uma solução inicial i (exemplo $i = x/2$)
2. Enquanto i^2 não for próximo o suficiente de x faça:
calcule novo i como $i = \frac{(i+x/i)}{2}$

Definimos que i^2 está próximo de x quando $|i^2 - x| < 0.1$

- ▶ Este algoritmo usa o conhecido **Método de Newton** e pode ser demonstrado matematicamente que converge para a raiz de x .

Exemplo - Problema

Vamos calcular a raiz quadrada de 9, usando o "Teste de Mesa":

1. $i = 4.5$, mas $i^2 = 20.25$ não está próximo de x , então atualizamos $i = (i + x/i)/2 = 3.25$
2. $i=3.25$, mas $i^2 = 10.56$ não está próximo de x , então atualizamos $i = (i + x/i)/2 = 3.0096$
3. $i=3.0096$ e $i^2 = 9.057$ estando próximo de x , portanto finalizamos o processo.

Achamos a solução aproximada 3.0096

Que veremos no curso...

- ▶ Aprenderemos principalmente a criar algoritmos para resolver pequenos problemas. Mas nada que dependa tanto de conhecimento matemático quanto o problema de calcular a raiz quadrada usando o método de Newton.
- ▶ Mas algum conhecimento de matemática talvez seja necessário em alguns problemas que veremos nesse curso.
- ▶ Um algoritmo pode ser descrito de várias formas, dentre elas, em português como vimos, com fluxograma ou em uma linguagem de programação.
- ▶ A vantagem de se implementar um algoritmo em uma linguagem de programação é que podemos a partir daí, criar um programa que usa o computador para resolver o problema.

Que NÃO veremos....

- ▶ Vocês NÃO vão aprender a usar programas neste curso (como office, MacOS, Linux, AutoCad, etc).
- ▶ Vocês VÃO ter porém, uma boa noção de como criar programas.
- ▶ Vocês NÃO vão aprender a fazer sites....enfim, vamos aprender as bases para poder criar todos os sistemas que vocês desejem.

Precisamos....

Para ir bem neste curso:

- ▶ Faça e implemente as listas de exercícios.
- ▶ E finalmente faça e implemente as listas de exercícios.

Horários e Avaliações

Ementa

Introdução

Resolução de problemas

Hardware e Software

Hardware

Software

O que é um computador?

- ▶ Um computador é uma máquina que, a partir de uma entrada, realiza um número muito grande de cálculos matemáticos e lógicos, gerando uma saída.
- ▶ Os computadores fazem isto muito bem e muito rápido.
- ▶ Computadores modernos fazem centenas de milhares de cálculos por segundo.
- ▶ Exemplo: Enquanto leio esta frase um computador típico executou mais de 1 Bilhão de instruções.

Hardware e Dispositivos

- ▶ Usualmente chamamos de Hardware todos os dispositivos físicos que compõem um computador, como CPU, Disco Rígido, Memória, etc.
- ▶ Estes dispositivos seguem uma organização básica denominada Arq. de Von Neumann.
- ▶ Ela é composta por três grandes pilares:
 - ▶ Unidade de Processamento Central: É a composição de registradores, da unidade de controle (UC), a unidade de lógica aritmética (ULA) e o contador de programa
 - ▶ Sistema de memória: Unidade que armazena os dados e os códigos do programa.
 - ▶ Sistema de entrada e saída: Unidade que é responsável por pegar os dados de entradas e mostrar os resultados de saída do programa.

Ciclo de execução de um programa

1. O Contador de Programa é utilizado pela Unidade de Controle para determinar qual e onde está a próxima instrução.
2. A Unidade de Controle busca a instrução do programa na Memória Principal.
3. Será feita a decodificação da instrução para uma linguagem na qual a ULA possa interpretar
4. Os dados requeridos são transferidos da memória e alocados nos registradores da CPU.
5. A Unidade de Lógica Aritmética executa a instrução e coloca os resultados na memória ou nos próprios registradores.

Figura: Arquitetura Von Neumann

O Gargalo de Von Neumann

- ▶ O gargalo de Von Neumann está no canal de transmissão entre a CPU e a memória, pois a memória não consegue trabalhar em frequências tão altas quanto a CPU, fazendo com que o CPU fique ocioso por um certo tempo.
- ▶ Alguns computadores ainda hoje utilizam a arquitetura antiga de processamento de instruções, como por exemplo, a calculadora de mesa. Que é feita somente para calcular números, não podendo instalar outro tipo de programa nela.

Resumo

- ▶ Todavia, os computadores Desktops, Notebooks, Smartphones, TVs etc utilizam de arquiteturas derivadas a de Von Neumann, todos eles possuem um CPU, Memória e I/O, mas com algumas diferenças como a separação da memória RAM e a do HD, uma para carregar os programas em execução e outra para salvar os dados e os programas, além de trabalhar em frequências altíssimas comparadas a da época.

Hardware e Dispositivos

- ▶ Todo o hardware opera com sinais com apenas 2 valores binários: representamos como 0 e 1
- ▶ A razão não é filosófica (“o computador só entende certo e errado”) mas eletrônica - é possível fazer circuitos que gastam muita pouca energia enquanto eles estão no estado 0 ou 1. Só gastam energia quando mudam de estado.
- ▶ A representação binária também facilita a interface entre o analógico (sinais magnéticos, elétricos, de luz, de rádio) e o digital - é mais insensível a ruídos

Hardware e Dispositivos

- ▶ A representação binária também facilita a interface entre o analógico (sinais magnéticos, elétricos, de luz, de rádio) e o digital - é mais insensível a ruídos
- ▶ Chamamos 1 desses sinais binários de bit
- ▶ Chamamos de Byte um agrupamento de 8 bits.
- ▶ Todas as informações armazenadas no computador são representadas por bits. Informações como letras, símbolos, imagens, programas são todas vários 0s e 1s.

Software

Software são os programas que executam tarefas utilizando o hardware de um computador.

- ▶ Os softwares são compostos por um conjunto de instruções que operam o hardware.
- ▶ Temos abaixo, por exemplo, três instruções para um computador de 32 bits.
- ▶ Um software é composto por milhares de instruções deste tipo:

```
0100 0010 0011 0101 0101 0100 0011 0110  
0100 1110 1100 1100 1001 0110 0110 1000
```

REFERÊNCIAS

- ▶ Robert W. Sebesta. Conceitos de Linguagens de Programação. Editora: Bookman. Edição: 11.
- ▶ ic.unicamp.br/wainer/cursos/1s2020/102/aula01.html