

Fundamentos de Lógica de Programação

Aula 1

Prof Tanilson Dias dos Santos

Universidade Aberta do Brasil – UAB
Universidade Federal do Tocantins - UFT

Quem sou eu?



- Graduação em Ciência da Computação - UFT;
- Mestrado em Sistemas e Computação - IME;
- Doutorado em Engenharia de Sistemas e Computação - COPPE/PESC - UFRJ;
- Currículo Lattes:
<http://lattes.cnpq.br/2812244463844431>
- Contato: tanilson.dias@uft.edu.br

Quem são vocês?



- **Vamos Criar uma mini-bio no Canva;**
- **Acessem o seguinte link:**
<https://encurtador.com.br/DGQS3>
- **Coloquem uma foto;**
- **Nome;**
- **Idade;**
- **Cidade;**
- **Uma curiosidade sobre você;**
- **O que você espera aprender nesta disciplina.**

Plano de Ensino

PLANO DE ENSINO



Plano de Disciplina - Fundamentos de Lógica de Programação

- **Ementa:** Fundamentos de computação relacionados à programação. Expressões aritméticas e comparativas. Operadores e expressões lógicas, tabelas verdade e propriedades. Variáveis e comando de atribuição. Entrada e saída. Estruturas de controle. Arranjos unidimensionais e bidimensionais. Modularização: funções e procedimentos.

Dinâmica da Disciplina

- 10 aulas síncronas/assíncronas;
- Ambiente Ava Moodle;
- Tutor On-line: Denis S. Passos



Aulas Online -
ConfWeb RNP

Tutores Presenciais

Polo	Tutor(a)
Ananás	Veruska
Araguaína	
Arraias	Deusmar
Gurupi	Itamar Júnior
Miracema	Juscimar

DATA	TÓPICO	TECNOLOGIA	HORÁRIO
20/05	Fluxogramas e Tabelas Verdade	Presencial/Gurupi	08h – 12h
10/06	Desvio Condicional e Comandos de Repetição	Presencial/Miracema	08h – 12h
01/07	Estruturas Complexas: Matrizes	Presencial/Arraias	08h – 12h

Forma de Avaliação

- 10 aulas, sendo a última Exame Final;
- Semanalmente, há uma atividade avaliativa, valendo 1 ponto, para os 9 encontros;
- A participação nos fóruns da disciplina vale 1 ponto;
- Se Nota Aluno (NA) maior que 7,0 APROVADO DIRETO;
- Se NA menor que 4,0 REPROVADO DIRETO;
- Se NA estiver entre 4,0 e 6,9, vai para Exame Final.
- Como funciona o Exame Final (EF) ?
- Uma prova valendo 10 pontos;
- Toma-se a Nota Final (NF):
$$NF = (NA + EF) / 2$$
- Se NF maior ou igual a 5 o aluno está APROVADO COM EXAME, caso contrário está REPROVADO COM EXAME.

Bibliografia e Metodologia

- **Programação Portugal:** Linguagem didática de português estruturado;
- **Referências:**
 - Livros e Apostilas sobre Fluxogramas, Tabelas Verdade, Algoritmos e Portugal;
 - Moraes, Paulo Sérgio. "**Curso Básico de Lógica de Programação.**" Unicamp-Centro de Computação-DSC (2000).
 - _____. "**Aprendizado de Algoritmos usando o Portugal Studio.**" UNIVALI – Universidade do Vale do Itajaí.
 - Coleção de Vídeos Sobre Portugal Studio.
 - Canal HM Programming:
https://www.youtube.com/watch?v=6OIADpFImtc&list=PLJ4lbwalqv3Eaiay2pCeU_QU6vb-Hz989&index=1



Teoria e Aplicações

- **Generalização e modelagem de problemas do mundo real;**
- **A disciplina de Fundamentos de Lógica de Programação se relaciona com outras disciplinas do curso:**
 - **Continuação imediata da disciplina Pensamento Computacional;**
 - **No próximo semestre: Programação de Computadores;**
 - **No 3º período: Algoritmos e Estruturas de Dados; Programação Orientada a Objetos;**
 - **Daqui pra frente, todos os períodos tem pelo menos uma disciplina de programação.**

Matriz Curricular Licenciatura em Computação

Figura 22 - Matriz Curricular do Curso de Licenciatura em Computação

1	Introdução ao EaD e ao AVA	Pensamento Computacional	Introdução à Computação	Fundamentos de Lógica de Programação	Comunicação Oral e Escrita	Prática de Ensino em Computação I
2	Fundamentos Filosóficos e Sociológicos da Educação	Educação e Tecnologias	Matemática Elementar e Discreta	Programação de Computadores	Arquitetura e Organização de Computadores	Prática de Ensino em Computação II
3	História da Educação e Política Educacional	Psicologia do Desenvolvimento e da Aprendizagem	Programação Orientada à Objetos	Algoritmos e Estrutura de Dados	Sistemas Operacionais	Prática de Ensino em Computação III
4	Gestão e Organização Escolar	Metodologia da Pesquisa Científica	Engenharia de Software	Banco de Dados	Montagem e Manutenção Computadores	Prática de Ensino em Computação IV
5	Educação: Didática, Planejamento e Avaliação	Probabilidade e Estatística	Análise e Projeto de Sistemas	Redes de Computadores	Êságio Supervisionado I	Prática de Ensino em Computação V
6	Metodologia do Ensino e Aprendizagem em Computação	Produção de Materiais Didáticos Digitais	Programação para Web	Inteligência Artificial aplicada à Educação	Êságio Supervisionado II	Prática de Ensino em Computação VI
7	Língua Brasileira de Sinais (LIBRAS)	Projeto de Tecnologias Aplicadas à Educação	Desenvolvimento de aplicativos aplicados à educação	Optativa I	Êságio Supervisionado III	Prática de Ensino em Computação VII
8	Trabalho de Conclusão de Curso - TCC	Optativa II	Empreendedorismo e Inovação	Aspectos Legais e Sociais da Computação	Êságio Supervisionado IV	Prática de Ensino em Computação VIII

BASE COMUM

CONTEÚDOS ESPECÍFICOS

PRÁTICAS PEDAGÓGICAS

PRÁTICAS EXTENSIONISTAS

OPTATIVAS



O que é um algoritmo?

- **Resposta mais comum:** Algoritmo é uma sequência finita de instruções executáveis que visam obter uma solução para um determinado tipo de problema;
- Nós utilizamos algoritmos no nosso dia-a-dia o tempo todo;
- Mas agora estamos interessados na execução de tarefas via computador.

Execução de Tarefas via Computador

- Para escrever um algoritmo em um computador é necessário pensar;
 - Determinar qual deve ser a SAÍDA;
 - Identificar quem são os dados de ENTRADA para se obter a saída;
 - Determinar como PROCESSAR a entrada para se obter a saída.





Vamos treinar?

- Identificar a **saída**, os elementos de **entrada** e o **processamento** para os seguintes problemas:
- Problema 1: Somar 2 números naturais;
- Problema 2: Encontrar a média de idade de um grupo de 3 pessoas.



Ciclo de Desenvolvimento de um Programa

- Analisar o Problema;
- Planejar a Solução;
- Escolher as Ferramentas;
- Codificar;
- Testar e Corrigir Erros;
- Produzir Documentação



O que significa Analisar o Problema?

- Compreender o que seu programa deve fazer;
- Ter uma ideia clara de como os dados do programa devem ser fornecidos;
- Perceber como entrada e saída se relacionam.



Como Planejar uma Solução?

- Pensar em uma sequência lógica de passos para resolver o problema;
- Para pensar no algoritmo é importante conhecer as ferramentas que você tem à sua disposição: comandos de leitura, gravação, operações matemáticas, lógicas, desvios condicionais, loops, etc;
- Existem vários métodos para especificar um algoritmo:
 - Diagramas de Fluxo/Fluxogramas;
 - Pseudo-códigos;
 - Códigos em linguagem de programação, etc.
- O planejamento também envolve testes manuais com dados representativos.



Como escolher as ferramentas?

- Primeiro é preciso conhecer as ferramentas;
- Suponhamos que você vai escrever seu código na linguagem Portugol;
- Quais os editores de texto, IDEs ou compiladores de Portugol você conhece?
- Quais os benefícios e dificuldades relacionados ao uso de cada um?
- Teste as ferramentas antes de escolhê-las.



Codificar

- Agora sim, nós efetivamente vamos inserir nosso código no computador;
- Ao final, quando escrevermos todo nosso código, teremos escrito nosso primeiro programa.



Testar e Corrigir Erros

- Não se assustem: dificilmente seu código vai rodar de primeira;
- Ao escrever um programa, diversos erros podem surgir;
- Esses erros são de diversas origens, mas as principais são:
 - Erros de Sintaxe: decorrem do desenvolvedor não ter escrito o código de acordo com a gramática que descreve a linguagem. Ex. Parênteses mal formados, pontuação, sobrando colchetes, etc;
 - Erros Semânticos: nesse tipo de erro o código está tecnicamente correto, mas há algum equívoco. Ex. Tentar abrir um arquivo chamado "arquivo.txt" mas o arquivo não existe;



Testar e Corrigir Erros

- Esses erros são de diversas origens, mas as principais são:
 - Erros de Lógica: Esses são os mais difíceis de diagnosticar. Não é erro de sintaxe e nem de semântica, o código está tecnicamente correto, tudo que foi solicitado faz sentido, mas a saída é diferente da esperada.
Ex. Foi codificado um programa para somar alguns números, mas em algum lugar foi colocado um sinal de subtração no lugar do sinal de soma;
 - Erros do Ambiente: ocorrem por uma particularidade da IDE utilizada para desenvolvimento. Ex. Um código funciona perfeitamente em uma máquina X, mas ao mudar para a máquina Y em outra IDE ele apresenta problemas, porque uma variável não foi inicializada antes de ser utilizada. A IDE da máquina X por padrão inicializa essas variáveis com 0 ou vazio, ou o equivalente do tipo base, mas a máquina Y deixa a variável com um "lixo" de memória.



Produzir Documentação

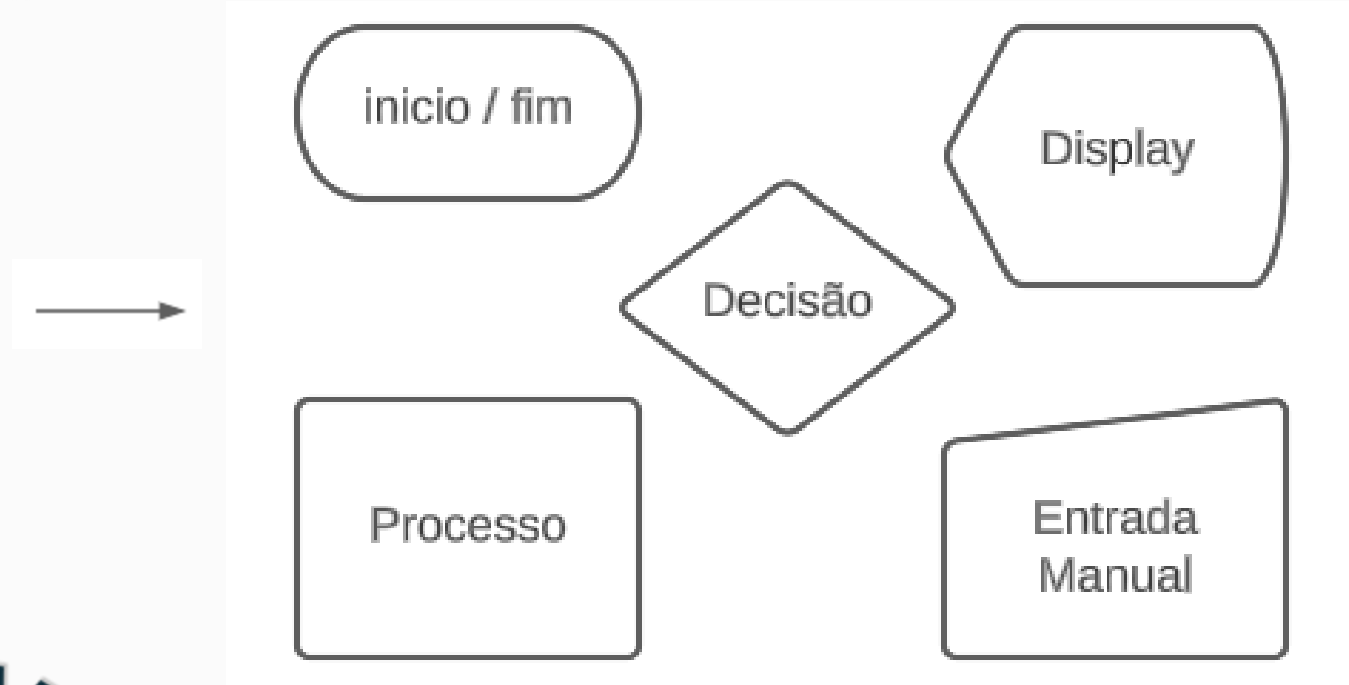
- **É importante produzir documentação:**
 - Explicando como seu código funciona;
 - Qual o objetivo do programa;
 - Como utilizar;
 - A função de cada variável, etc.
- Sua documentação vai servir como manual de instruções para entender o seu programa;
- A documentação também pode conter desenhos ou fluxogramas.



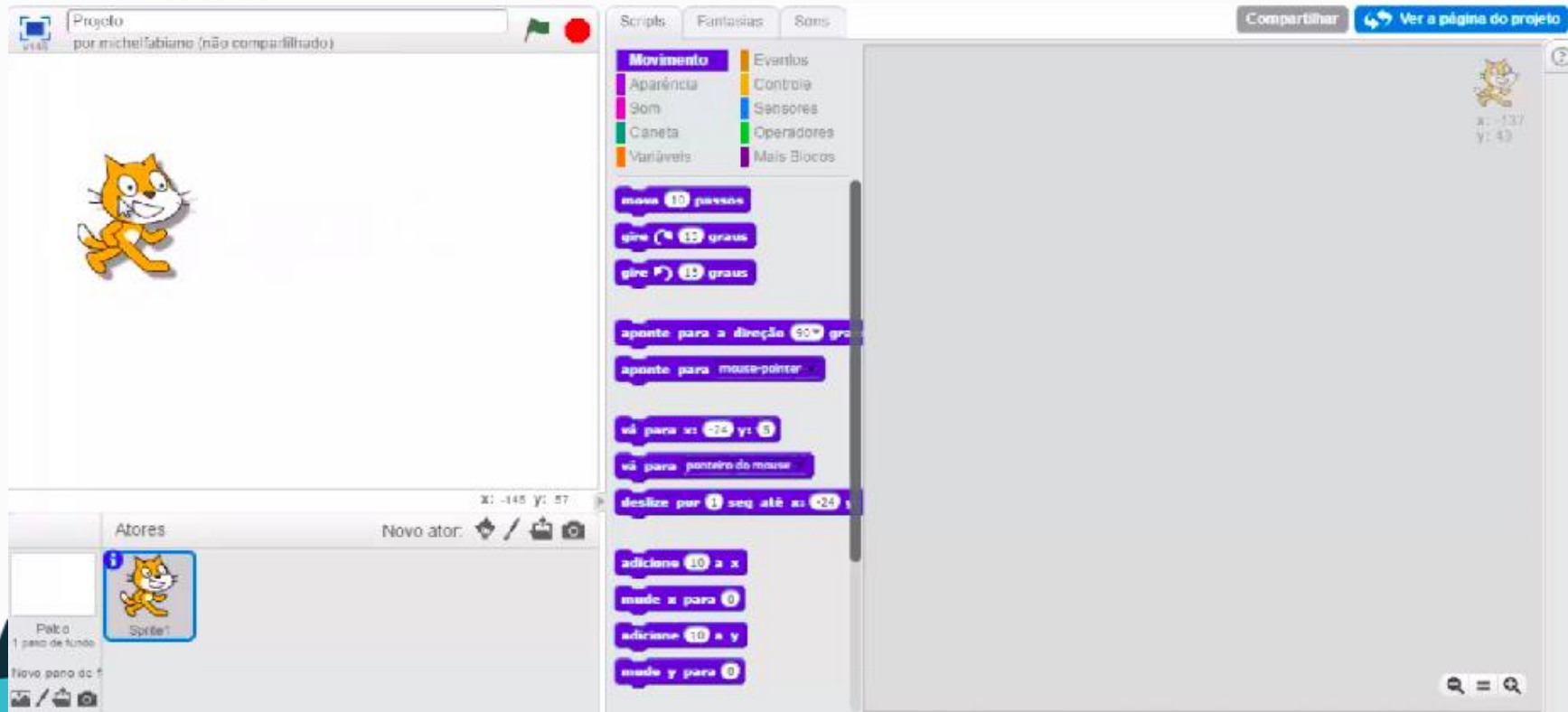
Fluxogramas ou Diagramas de Fluxo

- Descrevem graficamente a sequência de passos de um algoritmo;
- São construídos por um conjunto de símbolos geométricos, i.e. desenhos;
- Vamos utilizar símbolos do padrão ANSI do LucidChart.

Fluxogramas - Padrão ANSI / LucidChart



Lembram do Scratch ?





Lembram do Scratch?

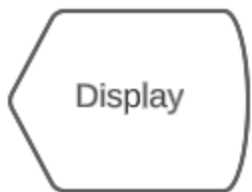
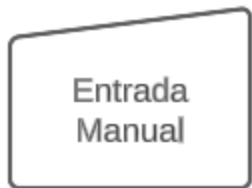
- Lá você definia um palco para apresentação do seu personagem;
- Vocês definiam comandos de repetição;
- Definem uma sequência de comandos;
- Definem o que ocorre quando um botão é apertado;
- Tomam decisões e mudam o ambiente quando uma situação acontece, etc.

Padrão Ansi / LucidChart



- Representa o **INICIO** e o **FIM** de um Algoritmo.
- Utilizado para operações que exigem **PROCESSAMENTO** de Dados ou Operações Aritiméticas.
- Efetua teste condicional, isto é, toma uma **DECISÃO**. Possui uma entrada, e duas saídas: VERDADEIRO E FALSO.

Padrão Ansi / LucidChart



- **LEITURA** de dados via teclado.
- Impressão de dados na tela, isto é **MOSTRAR** algum texto.
- Linha de **FLUXO**. Liga outros símbolos e indica a sequência de operações.



Exemplo 1

- Escrever o fluxograma para um programa que leia 2 números dados pelo usuário e apresente a soma dos 2 números.

Exemplo 2

- Escrever o fluxograma para um programa que leia do usuário a base e a altura do triângulo e calcule a área.
 - Área do Triângulo: $A = (b \cdot h) / 2$

Exemplo 3

- Escrever o fluxograma para um programa que leia a nota do aluno, dada pelo teclado, verifique se ele passou direto, reprovou direto ou foi para exame final.



Exemplo 4

- Escrever o fluxograma para um programa que leia o nome e a idade de 3 pessoas e diga o nome de qual dos 3 é o mais velho.
 - Considere que os 3 tem idades diferentes.



Tarefas Semanais

- Refazer Exercícios da Aula;
- Responder Questionário Avaliativo;
- Responder Fórum;
- Fazer leitura recomendada;
- Instalar compilador de Portugol (recomendo Portugol Studio);
- Estudar mais sobre Fluxogramas.



Conclusão e Próxima Aula

- **Aula de Hoje:**
 - Apresentação da Disciplina;
 - Conceitos Básicos de Lógica e Algoritmos;
 - Fluxogramas.
- **Próxima Aula:**
 - Exercícios sobre Fluxogramas;
 - Tabela Verdade.

Fundamentos de Lógica de Programação

Aula 1

Prof Tanilson Dias dos Santos

Universidade Aberta do Brasil – UAB
Universidade Federal do Tocantins - UFT