



Fundamentos de Programação



Agenda

- Quem sou eu? Quem são vocês?
- Como serão as aulas e avaliações?
- Aula
- Exercícios

Quem sou?

Quem sou?

Mestrado: Inteligência Artificial: *Sistemas Tutores Inteligentes e Computação Afetiva*

Histórico
Acadêmico

Diógenes Goldoni
UNISINOS

Nasci em 1991

Bacharel em Comunicação Digital

Bacharel em Ciência da
Computação

Mestre em Computação Aplicada
Inteligência Artificial

Quem sou?

Mestrado: Inteligência Artificial: *Sistemas Tutores Inteligentes e Computação Afetiva*

Profissional: Trabalho na área desde 2008. *Coordenador de Ciência de Dados no Sicredi*

Histórico
Acadêmico

Diógenes Goldoni
UNISINOS | SICREDI

Histórico
Profissional

Nasci em 1991

Grupo RBS

Bacharel em Comunicação Digital

Renner

Bacharel em Ciência da
Computação

Dell

Mestre em Computação Aplicada
Inteligência Artificial

Sicredi & Unisinos

Quem são
vocês?

A disciplina

A disciplina

Teoria...

A disciplina

Teoria...Prática...

A disciplina

Teoria...Prática...e mais prática!

A disciplina

Teoria...Prática...e mais prática!

Linguagem: Java

Dados e seus processadores

Dados e seus processadores

- **Dados:** Algum elemento que possa ser processado
- **Tipos**
 - Numérico: 9, 2, 13...
 - Alfabético: Letras (A-Z)
 - Alfanumérico: Números, letras e caracteres especiais ({, [, >, #, +, /)

Dados e seus processadores

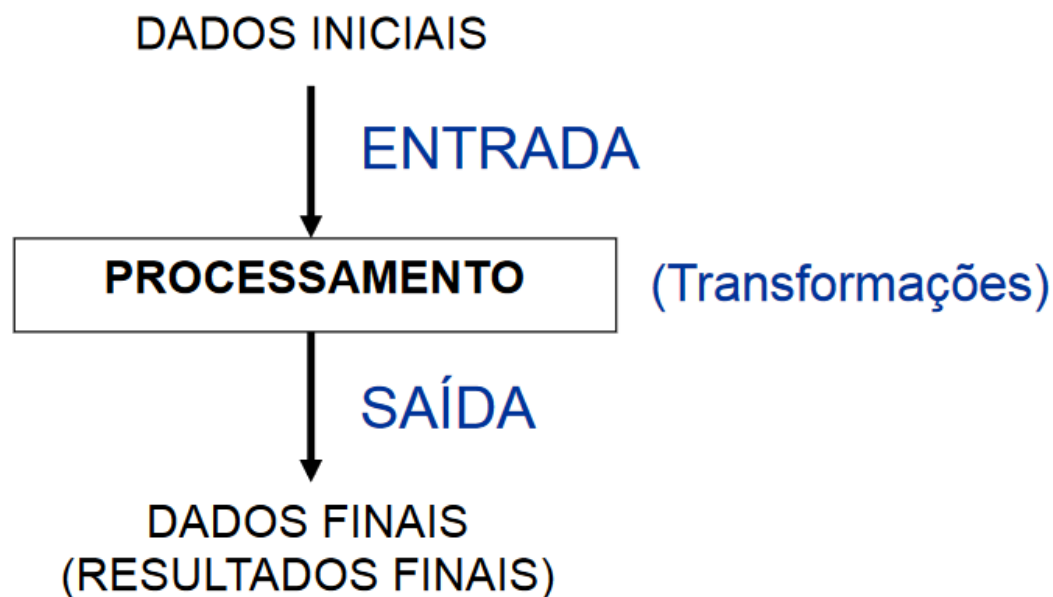
Processamento de Dados:

- Processo de **receber** dados, **processá-los** e produzir novos dados ou **resultados**

Dados e seus processadores

Processamento de Dados:

- Processo de **receber** dados, **processá-los** e produzir novos dados ou **resultados**



Dados e seus processadores

Processamento de Dados:

- Processo de **receber** dados, **processá-los** e produzir novos dados ou **resultados**

Processamento manual:

- Procurar uma frase específica em um livro extenso e anotá-la em um caderno
- Somar valores de compras no supermercado

Dados e seus processadores

Processamento de Dados:

- Processo de **receber** dados, **processá-los** e produzir novos dados ou **resultados**

Processamento manual:

- Procurar uma frase específica em um livro extenso e anotá-la em um caderno
- Somar valores de compras no supermercado

Somos preguiçosos. E inteligentes!

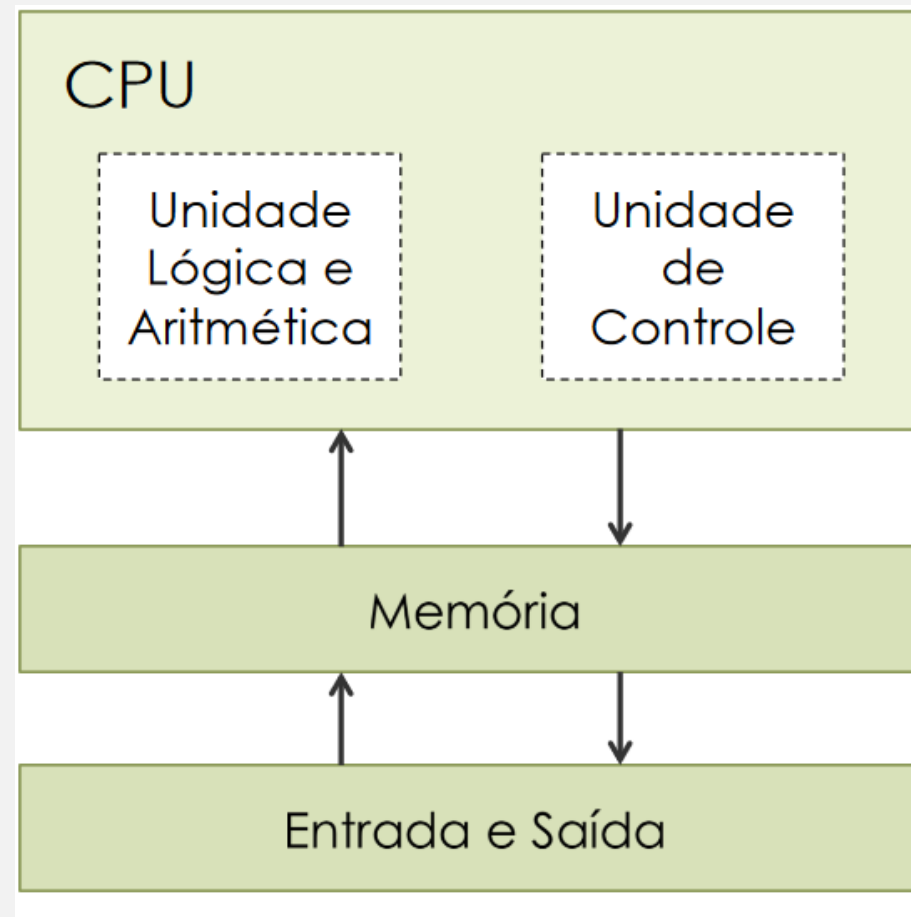
- Precisamos de algo que processe dados em **menor tempo** e com **mais eficiência...**
- Computadores!

Computadores e sua arquitetura

Arquitetura de Von Neumann

- A computação evolui a passos largos
- Ainda assim, a maioria dos computadores ainda segue a mesma arquitetura
 - Arquitetura de Von Neumann, proposta na década de 40.
- A arquitetura de Von Neumann é composta por 3 subsistemas:
 - Unidade Central de Processamento (CPU – Central Processment Unit)
 - Unidade de Memória Principal (RAM – Random Access Memory)
 - Unidades de Entrada e Saída (I/O – Input and Output)

Arquitetura de Von Neumann



Arquitetura de Von Neumann

- Unidade Central de Processamento (CPU)
 - Unidade de Controle: **Identifica** e **decodifica** instruções, acionando a unidade responsável
 - Unidade Lógica e Aritmética (ULA): Executa operações **lógicas** e **aritméticas**
- Memória Principal (RAM)
 - Armazena **instruções** e **dados** de um programa para garantir sua execução eficiente
- Unidades de Entrada e Saída (I/O):
 - Entrada: Transfere informação externa à máquina para a memória (mouse, teclado, etc.)
 - Saída: Exibe as informações guardadas na memória (monitor, caixas de som, etc.)

Arquitetura de Von Neumann

- Unidade Central de Processamento (CPU)
 - Unidade de Controle: **Identifica** e **decodifica** **instruções**, acionando a unidade responsável
 - Unidade Lógica e Aritmética (ULA): Executa operações **lógicas** e **aritméticas**
- Memória Principal (RAM)
 - Armazena **instruções** e **dados** de um programa para garantir sua execução eficiente
- Unidades de Entrada e Saída (I/O):
 - Entrada: Transfere informação externa à máquina para a memória (mouse, teclado, etc.)
 - Saída: Exibe as informações guardadas na memória (monitor, caixas de som, etc.)

→ O que é uma instrução?

Instruções

- Instrução

- Comando indicando ao computador quais os dados a serem operados, e operações a serem executadas.

- Exemplo:

7	−	5	=
---	---	---	---

- A CPU irá executar uma série de passos para que cada uma dessas instruções sejam efetuadas

Instruções – Etapas de processamento

- Unidade de Controle busca a instrução na memória
 - Identifica e decodifica a instrução, definindo as posições de memória necessárias para a execução
 - Transfere os dados para a Unidade Lógica e Aritmética
- Unidade Lógica e Aritmética executa as operações sobre os dados
 - Envia o resultado obtido para a memória

Linguagem de Máquina

VS

Linguagem de Alto Nível

Linguagem de Máquina e de Alto Nível

- Podemos passar ao computador um conjunto de instruções que desejamos que sejam executadas!
- Porém...o computador não entende a nossa linguagem
- O computador entende apenas 0 e 1
- Difícil programar em 0 e 1
- Solução:
 - Escrever programas em linguagem de alto nível (Python, Java, C, C++, etc)
 - Deixamos que **interpretadores** e **compiladores** traduzir nosso programa para linguagem de máquina

Linguagem de Máquina e de Alto Nível

Exemplo de Código em Assembly:

```
dint
setf      16,1,0
setf      32,1,1
.if      printer
move      sp,@laststack,1
move      a0,@laststa0,1
.endinf
movi      stckst,sp,1
```

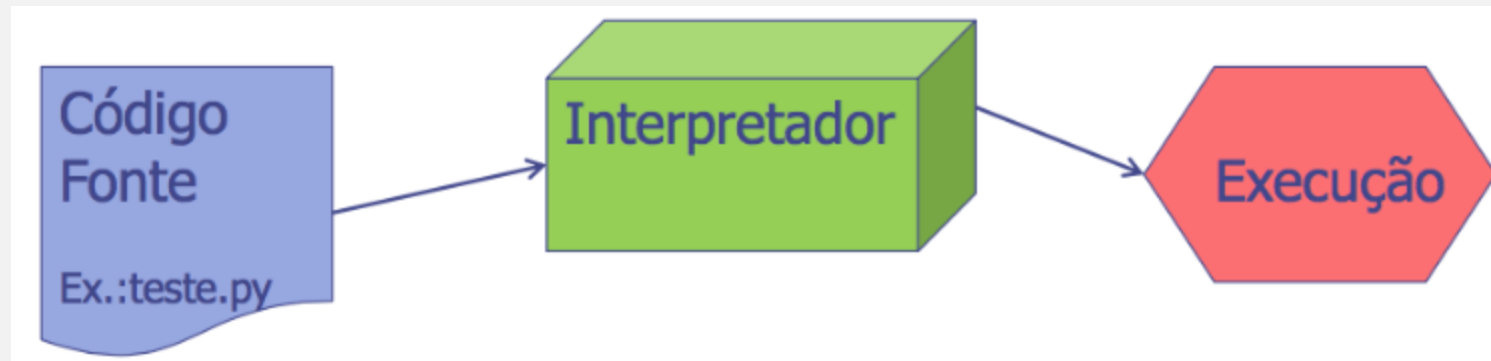
PERGUNTA:

Assembly é linguagem de máquina?

Linguagem de Máquina e de Alto Nível

Processos de interpretação de programas

- Um interpretador é um programa que recebe como entrada um arquivo contendo código fonte
- O interpretador lê linha a linha este arquivo, e executa **uma a uma** as suas instruções



Linguagem de Máquina e de Alto Nível

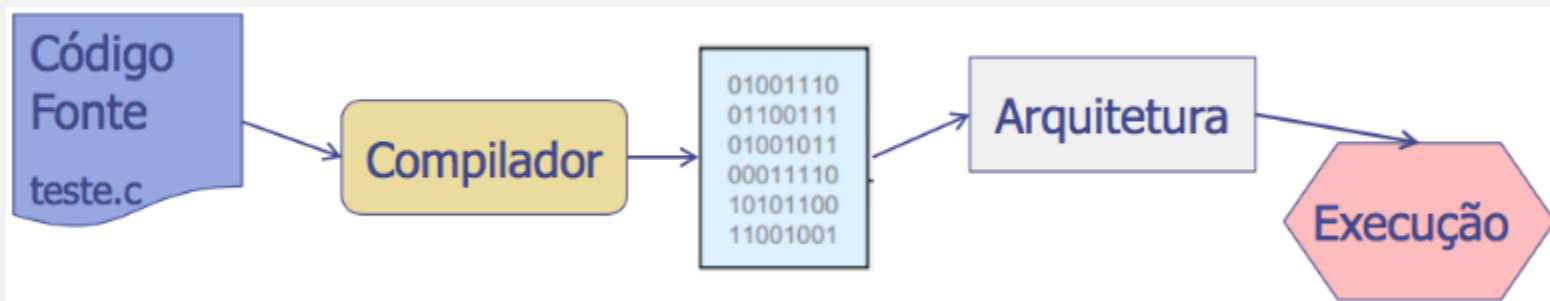
Processos de interpretação de programas

- Um programa pode ser executado em qualquer máquina, desde que haja um interpretador
- Programas interpretados tendem a ser mais lentos, pois cada linha precisa ser lida e decodificada
- Se houver um erro no código, o programa será interrompido abruptamente

Linguagem de Máquina e de Alto Nível

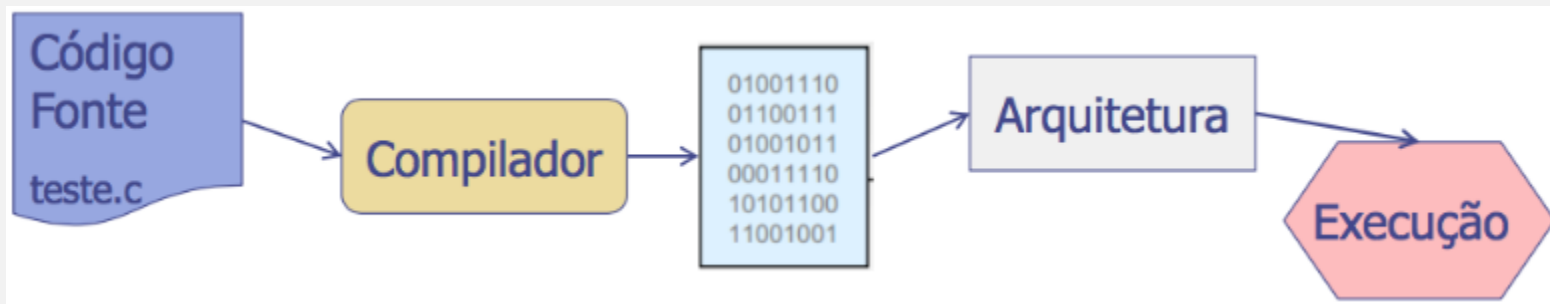
Processos de compilação de programas

- Um compilador recebe como **entrada** um arquivo de código fonte e cria um novo arquivo de **saída**
- O arquivo de **entrada** é o código fonte escrito por nós, programadores
- O arquivo de **saída** produzido pelo compilador, é normalmente identificado como **código objeto**.
- Este **código objeto** contém instruções de baixo nível, traduzidas para a linguagem de máquina!



Linguagem de Máquina e de Alto Nível

Processos de compilação de programas



- O **código objeto** *não* é portátil para qualquer arquitetura
- Diferentes compiladores são construídos para **diferentes arquiteturas** de processadores.
- O compilador só gera corretamente o código objeto se não encontrou **erros de compilação**

Algoritmos

O que são?

Algoritmo: O que são?

- Uma **sequência de passos** que devem ser seguidos para alcançar um objetivo
- Exemplos de algoritmos do **dia-a-dia**:
 - Vir para a Unisinos
 - Escovar os dentes
 - Comprar comida

Algoritmo: Formas de representação de um algoritmo

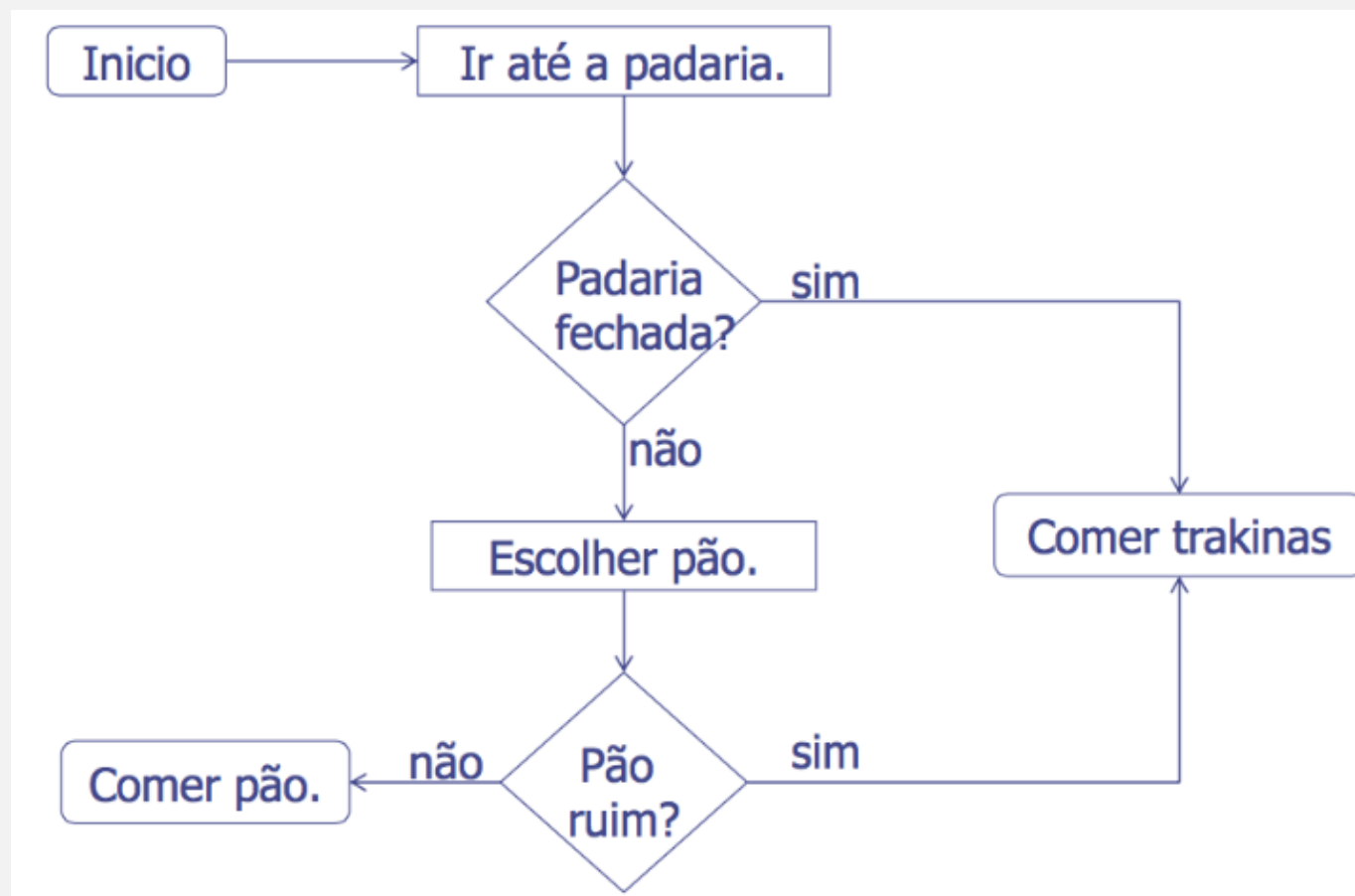
- Representamos algoritmos em nível lógico
 - Abstraímos detalhes referentes ao código e linguagens específicas
- Principais formas de representar algoritmos
 - Descritiva (narração)
 - Fluxograma
 - Pseudocódigo (português estruturado)

Algoritmo: Representação descritiva (narração)

- Algoritmos descritos diretamente em uma linguagem natural
- Exemplo: Algoritmo para fazer Massa Miojo
 - Coloque água em uma panela
 - Ferva a água
 - Abra o pacote de miojo
 - Quando a água estiver fervendo, insira o Miojo na panela
 - Espere 3 minutos
 - Desligue o fogo
 - Abra o pacote de tempero
 - Coloque o tempero e sirva

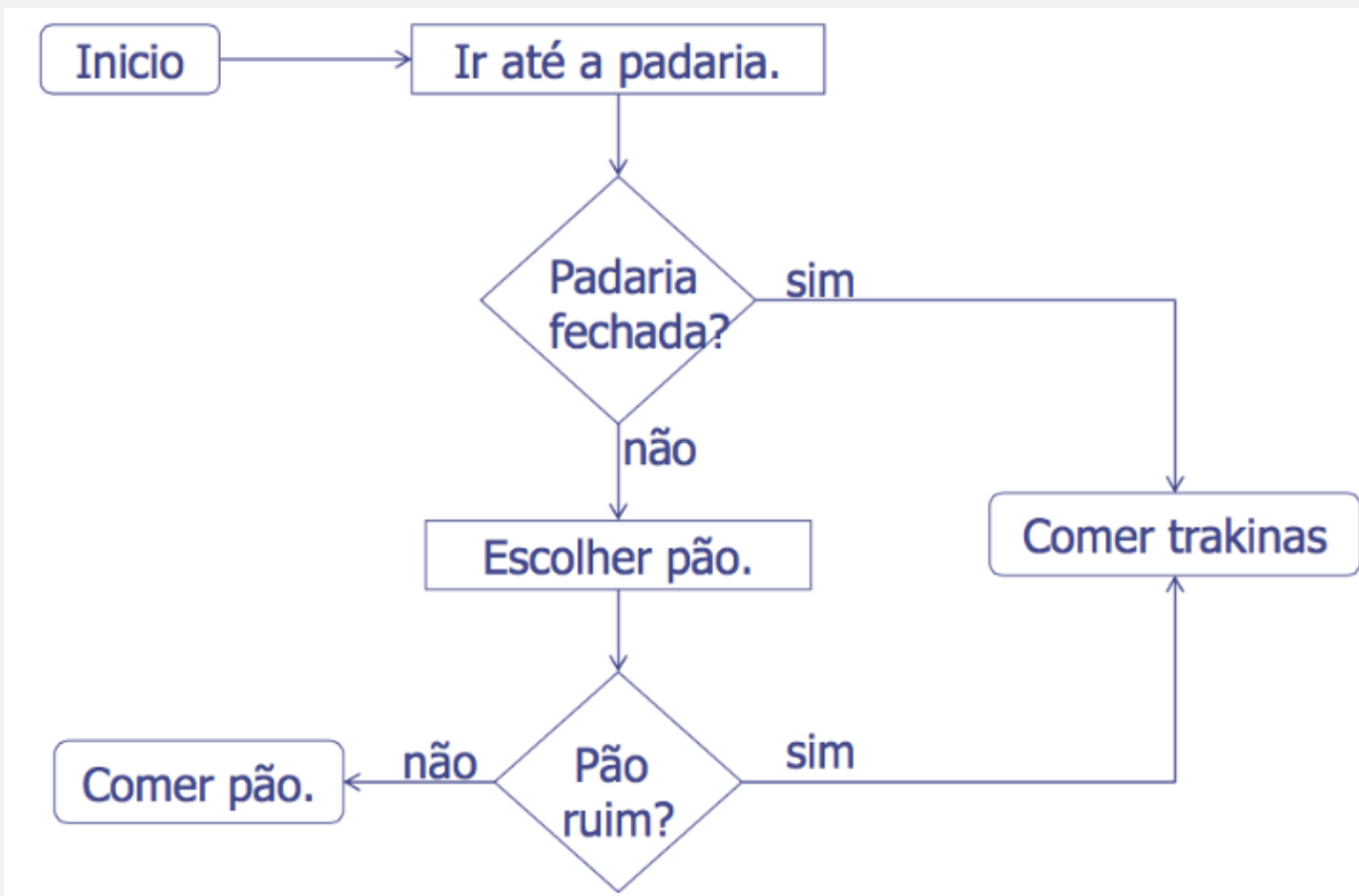
Algoritmo: Representação via fluxograma



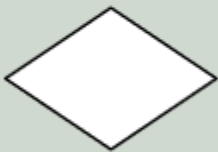

- Um **fluxograma** representa um algoritmo de forma gráfica



Algoritmo: Representação via fluxograma

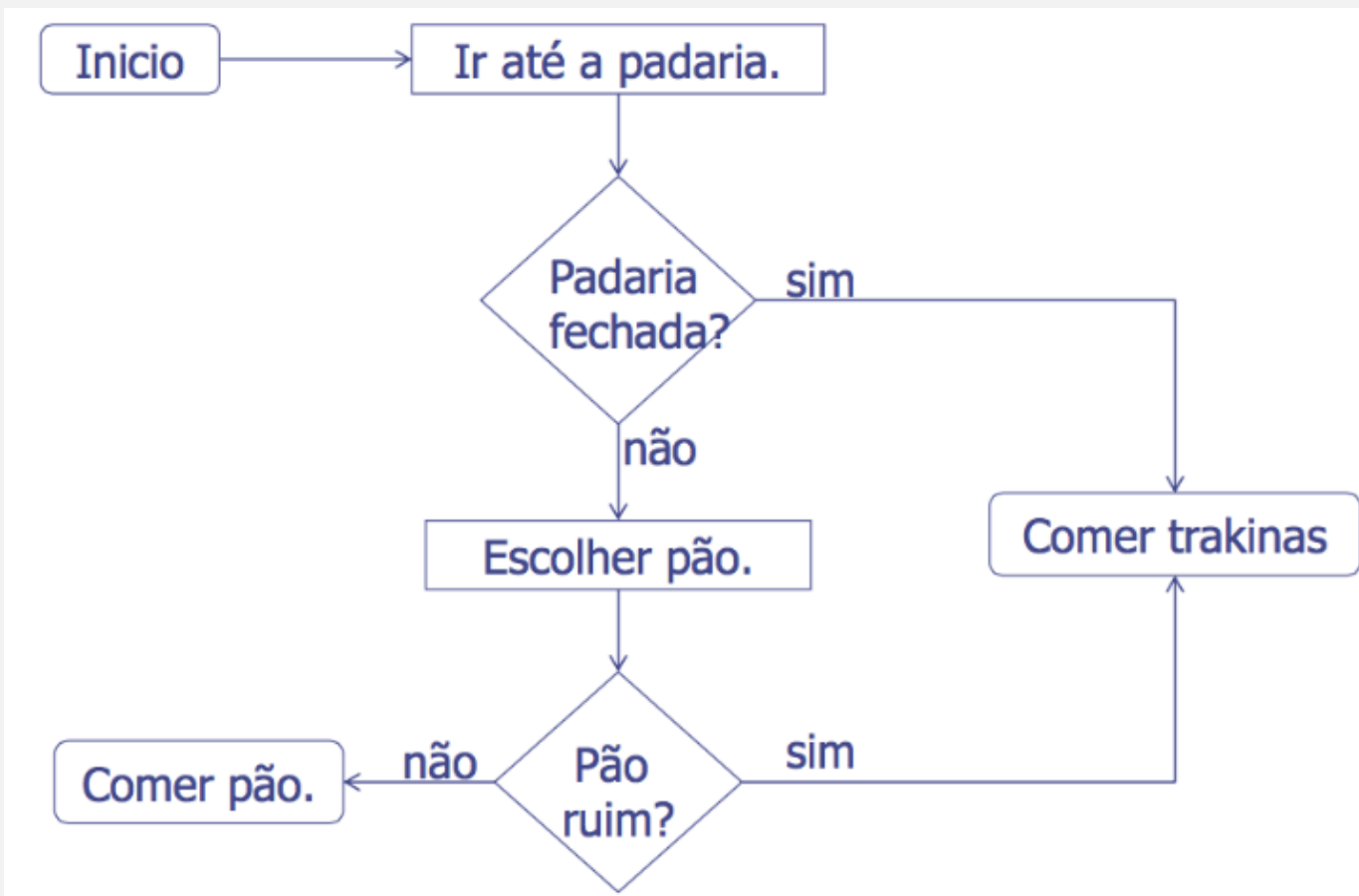
- Principais símbolos utilizados em fluxograma


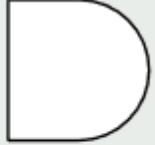



Símbolo	Significado
	Terminador. Indica início e fim do fluxograma.
	Processo. Indica uma determinada ação.
	Decisão. Indica que uma ação deve ser tomada em caso positivo (sim) ou negativo (não).
	Entrada manual. Representa uma entrada de dados pelo teclado.

Algoritmo: Representação via fluxograma

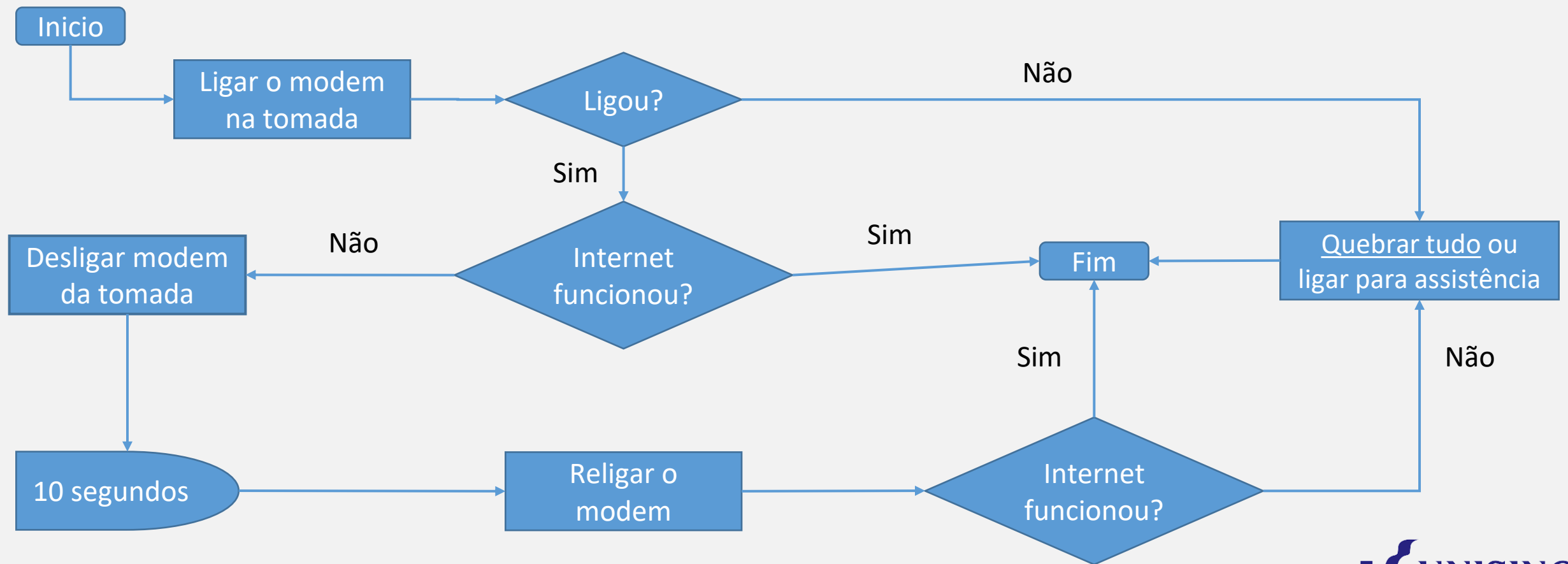
- Principais símbolos utilizados em fluxograma



Símbolo	Significado
	Fluxo. Indica o fluxo (direção) das ações.
	Atraso. Indica um determinado tempo sem ações.
	Exibição. Símbolo usado para mostrar dados na tela ou representar uma impressão (em impressora).

Algoritmo: Representação via fluxograma

- Algoritmo em fluxograma para ligar a internet em casa



Algoritmo: Representação via pseudocódigo

- Mostra o algoritmo de forma mais estruturada, mas em linguagem natural (português, inglês, etc)
- Exemplo: Algoritmo para calcular a média das 3 notas de um aluno
 - Solicitar que o usuário digite no teclado a **nota 1** do aluno, e armazenar em **NOTA_1**
 - Solicitar que o usuário digite no teclado a **nota 2** do aluno, e armazenar em **NOTA_2**
 - Solicitar que o usuário digite no teclado a **nota 3** do aluno, e armazenar em **NOTA_3**
 - Somar $NOTA_1 + NOTA_2 + NOTA_3$ e armazenar o resultado em **SOMA**
 - Dividir **SOMA** por 3 e armazenar o resultado em **MEDIA**
 - Imprimir na tela o valor de **MEDIA**

Exercícios para entregar

- Crie algoritmos via **fluxograma** e entregue via Moodle até a próxima aula
 - Crie um algoritmo que solicita nome do usuário pelo teclado e imprime na tela este nome
 - Crie um algoritmo que solicita o nome e idade de dois jogadores (um de cada vez). Ao final, imprima na tela o **nome do jogador mais velho**.

PERGUNTAS?