



# *Fundamentos de Programação*

---

Profa Melise Maria Veiga de Paula  
[melise@unifei.edu.br](mailto:melise@unifei.edu.br)

Universidade Federal de Itajubá

# Organização de Computadores

---

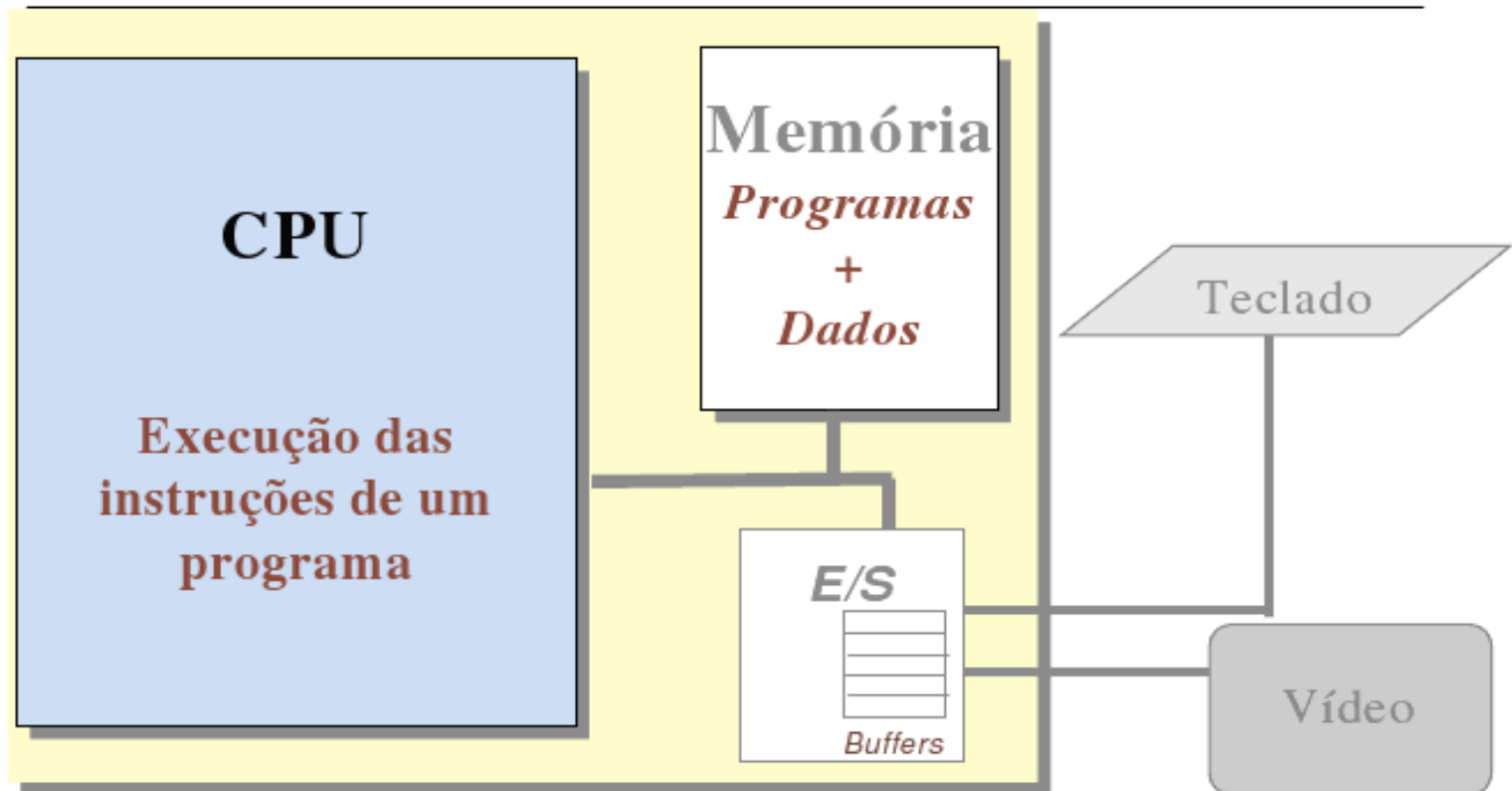
## ○ Computador

- Máquina capaz de resolver problemas através da execução de instruções

## ○ Vantagens:

- Capacidade de resolver rapidamente alguns problemas
- Capacidade de armazenamento

# Computador Digital



# Algoritmos x Programa

---

- Um programa é a formalização de um algoritmo em uma determinada linguagem de programação
- O programa é responsável por determinar o que deve ser feito pelo hardware
- Para isso, o programa deve ser escrito em uma linguagem que o computador entende

# Algoritmos x Programa

---

- Etapas para desenvolvimento de um programa
  - Analisar o problema
    - Identificar os dados disponíveis e qual é a informação que deve ser criada
  - Projeto a solução
    - Definir como a solução do problema será construída ou implementa. Qual é a melhor estratégia para resolver o problema analisado?
  - Implementação do programa
    - Codificar a solução em uma Linguagem de Programação
  - Validação do programa (teste)

# Algoritmos x Programa

---

- Três aspectos são importantes em um Algoritmo Computacional
  - (1) Técnica escolhida para programar
    - Forma como o programa será entendido pelo programador
    - Define o estilo de programação (uma história pode ser escrita de diferentes formas (narrativa, descritiva, etc ...))
    - Para programação, estas formas são denominadas paradigmas de programação
      - Programação Estruturada e Orientada a Objetos
    - No nosso caso, programação estruturada

# Algoritmos x Programa

---

- Três aspectos são importantes em um Algoritmo Computacional
  - (2) Estrutura de dados
    - Dados são manipulados durante a execução do programa na memória principal
    - Uma estrutura de dados define a forma como os dados são estruturados na memória para que sejam manipulados pelo programa
    - A escolha de uma estrutura interfere em vários aspectos
      - *Eficiência em relação ao tempo gasto para resolver um problema*
      - *Eficiência em relação aos recursos gastos para resolver um problema (memória)*

# Algoritmos x Programa

---

- Três aspectos são importantes em um Algoritmo Computacional
  - Linguagens de programação
    - Sistema de **codificação** constituído por conjuntos de **palavras**, cada qual com um só significado, formando séries de **instruções (comandos)** que devem ser **executadas** pelo **computador**
    - Seguem um determinado estilo (paradigma)
      - Linguagens Estruturadas (C, Pascal)
      - Orientadas a Objetos (C++, Java)
    - Podem ser classificadas em
      - Alto Nível
      - Baixo Nível
      - Máquina



# Algoritmos x Programação

---

- Linguagens de programação
  - Linguagens de Máquina (código de máquina)
    - É a linguagem entendida pelo computador
    - É denominada *linguagem binária* pois é constituída apenas dos dígitos *zero* e *um*
      - 0101 0011 1111 0001

# Algoritmos x Programação

---

- Linguagens de programação
  - Linguagens de baixo nível
    - Comandos substituem o código de máquina
    - Instruções são substituídas por mnemônicos
    - Assembly

## Código de máquina

## Assembly

```

00000000 011100000001
00000001 100100000011
00000010 000100000110
00000011 000000000000
00000100 000000000000
00000101 000000000000
00000110 010100000000
00000111 001000010101
00001000 100100000101
00001001 101100000000
00001010 001000010101
00001011 100100000100
00001100 111000000101
00001101 100100000101
00001110 100000000100
00001111 101100000000
00010000 001000010111
00010001 100100000100
00010010 111000000101
00010011 100100000101
00010100 000100001110
00010101 100000000011
...

```

```

ACCM 01x
ESCR NRUM
SLTA INIC

NRUM:
CONT:
PROD:
INIC:  ENTR
      SLTZ  UMZR
      ESCR  PROD
      DECR
      SLTZ  UMZR
      ESCR  CONT
      MULT  PROD
      ESCR  PROD
LOOP:  LEIA  CONT
      DECR
      SLTZ  ACAB
      ESCR  CONT
      MULT  PROD
      ESCR  PROD
      SLTA  LOOP
UMZR:  LEIA  NRUM
      SLTA  EXIB
ACAB:  LEIA  PROD
EXIB:  SAID
      ALTO
Listagem 2

```

# Algoritmos x Programação

---

- Linguagens de Programação
  - Linguagens de Alto Nível
    - Mais distantes das linguagem de máquina e mais próxima da linguagem usada por humanos para expressar problemas e algoritmos
    - Um comando em uma linguagem de alto nível tem sempre instruções equivalentes numa linguagem de baixo nível e em código de máquina
    - Exemplos:
      - Fortran, Cobol, Pascal, C, C++ e Java

# Algoritmos x Programação

---

- Problema básico

- Como fazer com que o computador entenda comandos de linguagens de alto nível?

- Tradutor

- Programa que traduz o código de um programa escrito em uma linguagem qualquer (código fonte) para um código equivalente escrito em outra linguagem (denominada código objeto)



## Código de máquina

```

00000000 011100000001
00000001 100100000011
00000010 000100000110
00000011 000000000000
00000100 000000000000
00000101 000000000000
00000110 010100000000
00000111 001000010101
00001000 100100000101
00001001 101100000000
00001010 001000010101
00001011 100100000100
00001100 111000000101
00001101 100100000101
00001110 100000000100
00001111 101100000000
00010000 001000010111
00010001 100100000100
00010010 111000000101
00010011 100100000101
00010100 000100001110
00010101 100000000011
...

```

## Assembler

## Assembly

```

ACCM 01x
ESCR NRUM
SLTA INIC

NRUM:
CONT:
PROD:
INIC: ENTR
      SLTZ UMZR
      ESCR PROD
      DECR
      SLTZ UMZR
      ESCR CONT
      MULT PROD
      ESCR PROD
LOOP: LEIA CONT
      DECR
      SLTZ ACAB
      ESCR CONT
      MULT PROD
      ESCR PROD
      SLTA LOOP
UMZR: LEIA NRUM
      SLTA EXIB
ACAB: LEIA PROD
EXIB: SAID
      ALTO
Listagem 2

```

# Algoritmos x Programas

---

## ○ Compilador

- Tradutor em que a linguagem fonte é uma linguagem de alto nível e a linguagem objeto é uma linguagem de baixo nível (assembly) ou máquina
- Efetua a tradução de todo o código-fonte em linguagem de alto nível para as instruções correspondentes em linguagem de máquina, gerando o código-objeto do programa.
- Em alguns casos, é necessário o uso de um outro programa (Link-Editor) que é responsável pela junção de diversos códigos-objeto em um único programa executável
- Pascal, C, C++

# Programas ... Alguns termos

---

- Compilar um programa

- Significa traduzir o programa para linguagem de máquina
- Caso o programa tenha algum erro de sintaxe da linguagem, ocorre um erro de compilação

- Executar o programa

- Depois de compilado, o programa pode ser executado (colocar em funcionamento)
- Erros nesta etapa são causados por problemas na lógica do programa



# Programas

---

- Um programa é basicamente um arquivo
- Dependendo do programa, além do arquivo correspondente, podem ser necessários, outros arquivos
- Mas o que é um arquivo????

# Arquivos

---

- É o local da memória da máquina onde as informações são armazenadas em pacotes ou grupos
- Pode ter qualquer tipo de informação: programas ou dados
- São identificados pelo nome e extensão
  - .c → arquivos de programas em linguagem C
  - .for → arquivos de programas em linguagem Fortran
  - .dat → armazena dados para um programa
  - .txt → armazena textos, que também podem ser dados de um programa
  - .java → armazena um programa em linguagem Java

# Algoritmos x Programação

---

- Ambiente Integrado de Desenvolvimento(Integrated Development Environment – IDE)
  - Um programa pode ser escrito em um editor de texto ou em alguma ferramenta específica
  - IDE (interface de desenvolvimento)
    - Ambiente de desenvolvimento que facilita a codificação
    - Não são linguagens de programação
    - Netbeans para Java
    - Dev C++ para C e C++
    - Eclipse

# Algoritmos x Programação

---

- Ambiente Integrado de Desenvolvimento(Integrated Development Environment – IDE)
  - Componentes
    - Editor de código
      - Facilitar a edição do código
    - Compilador
      - Traduz o código fonte para código objeto

# Algoritmos x Programação

---

- IDE

- Depurador

- Depurar: analisar o código para identificar a origem de erros e eliminá-los
    - Na depuração, podem ser definidos pontos de parada (breakpoints)
    - Em cada breakpoint, é possível verificar de que forma o programa está sendo executado. Por exemplo, valores assumidos por variáveis em um determinado momento.

- Linker (linkeditor-ligador)

- Carregador (Loader)