

*Professor Marco Câmara*

---

# Introdução à Computação para Engenharia.

UCSAL 2025-01

---

# Aula 01

---

# Por que estudar isso ?

---

Os dispositivos computacionais são parte de nossa vida

**Normalmente são mal utilizados:** poucos usuários utilizam todos os recursos.

*Estão em todos os lugares, inclusive nas obras e escritórios;*

Constituem uma ferramenta fundamental

*Armazenam, registram, calculam*



---

# Dispositivos Computacionais

---

## Tipos

*Eletrônica embarcada, microcontroladores, celulares, tablets, notebooks e desktops, servidores, , consoles de vídeo-game, computadores de grande porte;*

*Alguns não são acessíveis diretamente pelo usuário, outros exigem acesso especializado.*

## Características Comuns

*Processamento, armazenamento e comunicação digital e binária;*

*Sistemas Operacionais. Ex: Android, IOS, Windows, Linux, MacOS;*

*Unidade de armazenamento em bits e Bytes (8 bits) e seus múltiplos.*

Não temos mil, milhão, bilhão etc, e sim:

**1 nibble = 4 bits;**

**1 Byte = 8 bits;**

**1 KiloByte =  $2^{10}$  (1024) Bytes;** **1 MegaByte =  $2^{10}$  KB =  $2^{20}$  Bytes**

**1 GigaByte =  $2^{30}$  Bytes;**

**1 TeraByte =  $2^{40}$  KB**

**1 PetaByte =  $2^{50}$  Bytes;**

**[...]**

---

# Dispositivos Computacionais

---

## Armazenamento

**Armazenamento Interno** (memória RAM): espaço para execução dos processos (análogo à uma mesa?);

**Armazenamento Externo I** (CD/DVD (?), cartões de memória, pendrives, HDs e SSDs): espaço para armazenamento de informações permanentes, como programas e aplicativos (inclusive o Sistema Operacional) e arquivos diversos, com formatos definidos pelo programa ou aplicativo que o manipulará;

**Armazenamento Externo II** (outro dispositivo, servidor local, remoto e “nuvem”): função similar ao anterior, porém tem vantagens (acesso de qualquer lugar) e desvantagens (depende de recursos de comunicação para disponibilidade e performance).

---

# Dispositivos Computacionais

---

## Comunicação

**Rede "cabada" (ethernet):** suporta conexões a 10/100/1000 Mbps nominais, com baixa taxa de erro e alta estabilidade;

**Rede "wireless" (Wi-Fi):** suporta conexões a 11/54/108/400/1100Mbps nominais, com alta taxa de erro e nem sempre garante estabilidade;

**Bluetooth:** suporta conexões de periféricos externos a curta distância com taxa de transferência de 1/3/24/25 Mbps nominais, com alta taxa de erro e nem sempre garante estabilidade;

**3G/4G:** suporta conexões a operadora de telecomunicações com taxa de transferência variável com a localidade e distância da torre (até 2Mbps para 3G, até 300Mbps para 4G, e até 150Mbps para 5G nominais).

# Aula 02

---

# Tipos de *software*

---

## EPRROM/BIOS

*Vem gravada de forma não volátil; pode ser atualizada em procedimento específico;*

## Sistema Operacional

*Determina a operação do dispositivo e seus periféricos, a execução de aplicações e interface do usuário;*

## Aplicativos, Programas e Sistemas

*Executa atividades específicas para o qual foi programado; muitos admitem o desenvolvimento de ferramentas internas e até programação (scripts). Alguns podem ser executados remotamente (em um servidor, ou na “nuvem”), com o dispositivo do usuário funcionando apenas como tela e interface de entrada;*

## Linguagens de Programação e Ambientes de Desenvolvimento

*Ferramentas para o desenvolvimento de softwares, inclusive outras linguagens;*

# Aplicativos x Mundo Real

Os aplicativos substituem, em meio digital, procedimentos e dispositivos do mundo real:

- Processador de Texto;
- GPS e Mapas;
- Agendas e Calendários;
- Alguns de Fotos.

A planilha eletrônica é diferente ...



# Planilhas Eletrônicas

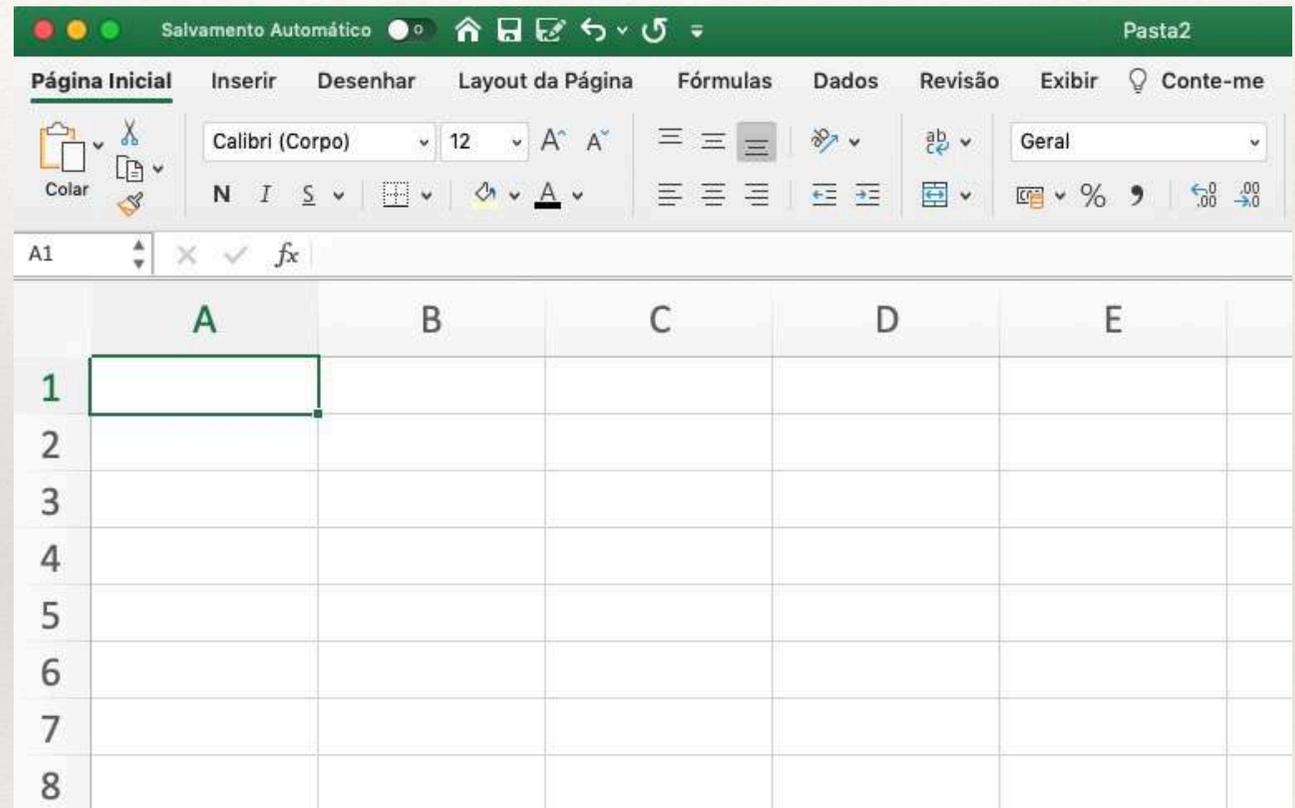
Podemos até comparar com uma tabela, mas ...

... e os tipos de célula?

... e os cálculos?

... e a formatação independente do conteúdo?

Uma planilha eletrônica não tem similar no mundo real.



# Histórico

**1979:** Visicalc no Apple II  
(responsável pelo sucesso da Apple?)

**1983:** Lotus 1-2-3 no IBM PC

**1989:** Borland Quattro

**1987:** Excel

**2006:** Google Docs (estréia das planilhas “na nuvem”)



C11 (L) TOTAL C1  
25

	A	B	C	D
	ITEM	NO.	UNIT	COST
	MUCK RAKE	4	12.95	556.00
	BUCK CUT	1	100.00	100.00
	TONER	250	49.95	12487.50
	EYE SNUFF	2	4.95	9.90
			SUBTOTAL	13155.50
			9.75% TAX	1282.66
			<b>TOTAL</b>	<b>14438.16</b>

---

# Versões em uso (2022)

---

## Microsoft Excel

Parte da suíte Office da Microsoft;

A maior base de usuários - sucesso inquestionável - “sinônimo de planilha”;

Admite licenciamento eterno, ou por assinatura (Office 365);

Possui versões para o Windows e MacOS que compartilham arquivos e possuem as mesmas funcionalidades;

Possui versão para iOS e Android, inclusive gratuitas (básicas) com funcionalidades limitadas;

Possui versão para nuvem, inclusive gratuita (básica), também limitada.

---

# Versões em uso (2022)

---

## Apple Numbers

Parte da suíte iWork da Apple;

Licenciamento gratuito, mas é compatível apenas com dispositivos Apple; versão limitada no iOS e na nuvem;

Muito forte nos recursos gráficos e visuais;

Compatível com o Excel na maior parte das funcionalidades.

---

# Versões em uso (2022)

---

## Libre Office

Licenciamento gratuito, é compatível com o Windows, MacOS, Linux e “na nuvem”;

Compatível com o Excel na maior parte das funcionalidades.

---

# Versões em uso (2022)

---

## Google Planilhas

Parte da suíte Google Docs;

Licenciamento gratuito;

Roda exclusivamente na nuvem;

Compatibilidade parcial com o Excel; funcionalidades limitadas.

---

# Planilha - Conceitos Básicos

---

Tabela de dimensões (quase) infinitas

- Linhas (numéricas);

- Colunas (alfabéticas);

- Abas (nomes);

- Arquivos (contém todas as abas).

Cada cruzamento de linha e coluna é chamado de **célula**, que pode conter:

- Um número;

- Um carácter numérico, alfabético ou contendo símbolos;

- Uma fórmula, que não é visível na tabela; o que se vê é o **resultado**;

- Qualquer conteúdo pode ser apresentado em diversos formatos (ou não).

---

# Planilha - Fórmulas

---

Variáveis tipicamente são células, ou grupo de células

As células referenciadas podem estar na mesma aba, em outra aba, ou mesmo em outro arquivo;

Algumas fórmulas retornam valores independente de variáveis. Ex: Data;

As células referenciadas podem conter qualquer conteúdo, inclusive outras fórmulas.

Fórmulas identificam referências por endereços

Referências podem ser relativos (o mais comuns) ou fixos;

A fixação de referências ocorre através do uso do carácter \$.

# Aula 03

# Uma Planilha Básica

## Conteúdo

Números

Textos

Fórmulas

## Referências

	Janeiro	%	Fevereiro	%	Março	%	
Condomínio	R\$ 500,00	29,9%	R\$ 510,00	25,4%	R\$ 525,00	28,0%	<b>R\$ 1.535,00</b>
Luz Elétrica	R\$ 350,00	21,0%	R\$ 600,00	29,9%	R\$ 400,00	21,3%	<b>R\$ 1.350,00</b>
Água	R\$ 120,00	7,2%	R\$ 130,00	6,5%	R\$ 120,00	6,4%	<b>R\$ 370,00</b>
Supermercado	R\$ 600,00	35,9%	R\$ 650,00	32,3%	R\$ 750,00	40,0%	<b>R\$ 2.000,00</b>
Outros	R\$ 100,00	6,0%	R\$ 120,00	6,0%	R\$ 80,00	4,3%	<b>R\$ 300,00</b>
	<b>R\$ 1.670,00</b>		<b>R\$ 2.010,00</b>		<b>R\$ 1.875,00</b>		<b>R\$ 5.555,00</b>

**Relativas**, ou seja, apontam a localização relativa dos argumentos;

Copiando fórmulas com referências relativas;

**Absolutas**, ou seja, identificam a coluna e a linha, impedindo a modificação.

## Formatação de Conteúdo

Formatos numéricos, alinhamento, cores, bordas, negrito etc

Montando um exemplo de orçamento doméstico (acima) no Google

Importação e Exportação para outras planilhas (Excel)

# Orçamento de Obra (prática)

## Criar planilha com abas

Abas contém conteúdos diferentes, porém relacionados.

## Preencher os dados “fixos”;

## Criar fórmulas nas abas

Na aba “Total”, usar fórmulas com referências às outras abas;

P.S. Exemplo do livro está errado.

## Salvar arquivo padrão;

Salvar arquivo específico, preencher quantitativos, e obter orçamento final;

Encaminhar impressão PDF para professor (avaliação diária);

### Solução:

O empreiteiro decide construir quatro planilhas-padrão no mesmo arquivo, uma para cada parte do orçamento. Ele planeja usar um arquivo para cada obra, e somente atualizar os campos com as novas quantidades para cada item, alterando as quatro planilhas-padrão para cada nova obra orçada. A situação pode ser visualizada nas Figuras numeradas de 1.22 a 1.25.

A quantidade de itens foi reduzida por motivos didáticos e tem apenas fim ilustrativo.

Item	Descrição	Unidade	Preço Unitário	Quantidade	Preço Total
1	Tube PVC DN 50	Barra	R\$ 14,50		
2	Joelho soldavel DN50	Pç	R\$ 0,80		
3	Te soldavel DN50	Pç	R\$ 1,00		
4	Registro esfera DN50	Pç	R\$ 3,20		
				<b>Total</b>	

Figura 1.22: Planilha-padrão para o orçamento Hidrossanitário

Item	Descrição	Unidade	Preço Unitário	Quantidade	Preço Total
1	Fio PVC 700C 2.5mm2	Role	R\$ 45,00		
2	Eletroduto flexivel	Role	R\$ 67,50		
3	Interruptor simples	Pç	R\$ 5,30		
4	Tomada 2P+N	Pç	R\$ 4,50		
				<b>Total</b>	

Figura 1.23: Planilha-padrão para o orçamento Elétrico

Item	Descrição	Unidade	Preço Unitário[CUB]	Preço Unitário[R\$]	Preço Total [R\$]
1	Projeto Elétrico	m2	0,0120		
2	Projeto Hidrossanitário	m2	0,0100		
3	Implantação	m2	0,0320		

Figura 1.24: Planilha-padrão para o orçamento de Serviços

Item	Descrição	Unidade	Preço Unitário	Quantidade	Preço Total
1	Tube PVC DN 50	Barra	R\$ 14,50	7	R\$ 101,50
2	Joelho soldavel DN50	Pç	R\$ 0,80	23	R\$ 18,40
3	Te soldavel DN50	Pç	R\$ 1,00	12	R\$ 12,00
4	Registro esfera DN50	Pç	R\$ 3,20	8	R\$ 25,60
				<b>Total</b>	<b>R\$ 157,50</b>

Figura 1.25: Planilha do Orçamento total

# Aula 04

# Sistemas de Equações Lineares (*prática*)

Matrizes são úteis na resolução de problemas de engenharia (treliças, circuitos elétricos etc);

Funções com Matrizes em uma planilha eletrônica

Formato de tabela denuncia compatibilidade;

Cálculo de soma, multiplicação, inversão e determinante de matrizes.

Matrizes são utilizadas em diversos métodos para resolução de Sistemas de Equações Lineares (Gauss, Sarrus etc)

Testando um exemplo

Calcula-se o determinante da matriz do sistema

$D = 0 \Rightarrow$  não é possível resolver

Determina-se a matriz inversa

Multiplicamos a matriz inversa pelo vetor dos termos independentes

$$3x + 4y + 2z = 2$$

$$3x - 6y + 2z = 4$$

$$2x + 2y - 4z = 7$$

$$\begin{bmatrix} 3 & 4 & 2 \\ 3 & -6 & 2 \\ 2 & 2 & -4 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 7 \end{bmatrix}$$

$$A \cdot B = C$$

$$B = C \cdot A^{-1}$$

---

# Mais um exemplo

---

$$2x - 4y + 6z = 3$$

$$3x + 5y - 2z - t = 22$$

$$-5x + 4y + 3z + 7t = 75$$

$$2x + 3y + 2z - t = 10$$

# Aula 05

---

# Análise de Dados

*(prática)*

---

Usando o Excel como SGBD (Sistema Gerenciador de Banco de Dados)

Fórmulas Condicionais

Ordenação

Filtros

Subtotais

Consulta a Tabelas Verticais

Formatação Condicional

---

# Subtotais

---

## Função SUBTOTAL

Retorna resultados baseados em um grupo de células, podendo considerar o estado do filtro.

Sintaxe: SUBTOTAL(código\_função, bloco1, [bloco2, ...])

**Código de Função:** 1 para Média (AVERAGE), 2 para Contagem Numérica (COUNT), 3 para Contagem Alfanumérica (COUNTA), 4 para Valor Máximo (MAX), 5 para Valor Mínimo (MIN), 6 para Produto dos itens (PRODUCT), 7 para Desvio Padrão (STDEV), 8 para Desvio Padrão de toda a população (STDEVP), 9 para Soma (SUM), 10 para Variância (VAR) e 11 para Variância de toda população (VARP).

Valores Ocultos podem ser desprezados se qualquer um dos códigos for precedido por 10 para códigos de um dígito, ou simplesmente 1 para códigos de 2 dígitos. Ex: 102 para contar desprezando valores ocultos.

# Aula 07

---

# Consulta de Cotações c/ Filtros e Macros

## *(prática)*

---

Usando o Excel como SGBD (Sistema Gerenciador de Banco de Dados)

Exemplo: Consulta de Cotações por Produto (planilha **Filtros com Macros v1.xls**)

Revisão de busca em tabela, desta vez em outra aba

Tabela de Itens com chave primária (código de 3 dígitos);

Separação em outra aba promove criação de “banco de dados”;

É possível consultar outro arquivo! (**Planilha externa consulta base.xls**)

Proteção de Células

Nem todas as células devem aceitar entradas pelo usuário;

Controle evita danos a planilhas “modelo”.

Revisão de Filtros

Gravação de Macros;

Atribuição de Macros a botões.

---

# Consulta de Cotações c/ Filtros e Macros II

## *(prática)*

---

### Revisão de Filtros

Seleção e ordenação por coluna.

### Gravação de Macros

Automação de atividades repetitivas;

Exemplo com atribuição de Filtro específico;

Habilitação do menu “Desenvolvedor”;

Gravação de processos repetitivos.

### Atribuição de Macros a botões

Macros executadas de 3 formas: no menu, com atalho de teclado, e através de botão;

Atribuindo botões a macros diferentes.

# Aula 08

---

# Introdução ao Scilab

---

**Scientific Laboratory** - *software* criado para resolução de cálculos numéricos, tanto interativos quanto através de programação;

Concorrente *free e open-source* do MathLab, que é proprietário.



Possui recursos avançados de visualização gráfica;

Mantido atualmente pela SciLab Enterprises;

Desenvolvimento começou em 1999, na França, no INRIA (*Institut National de Recherche en Informatique et en Automatique*), e depois administrado por um consórcio criado em 2003.

---

# Como instalar?

---

Está disponível no site [scilab.org](https://www.scilab.org)

Tem versões para Windows (Vista, 7, 8, 10), GNU/Linux e MacOS (Intel e M1);

Versão atual (Abr/2023): 2023.0.0;

Licença GPL (*General Public License*): basicamente dá o direito aos usuários de executar, estudar, compartilhar e modificar o *software*.

# Interface com o Usuário

The image shows the Scilab 6.1.1 user interface with several components highlighted by red boxes and labeled with red text and arrows:

- Navegador de arquivos**: Points to the file browser on the left side of the window.
- Console**: Points to the central command execution area.
- Navegador de Variáveis**: Points to the variable browser on the right side of the window.
- Histórico de Comandos**: Points to the command history panel at the bottom right.

The interface includes a top toolbar, a title bar "Scilab 6.1.1 Console", and a news feed at the bottom right with the following text:

**Scilab 6.1.1 has been released!!!**

Dear fellow users,

We have the pleasure to announce the release of the new version of Scilab. Check [here](#) to download and find more details about Scilab 6.1.1.

---

# *Prompt* de comando

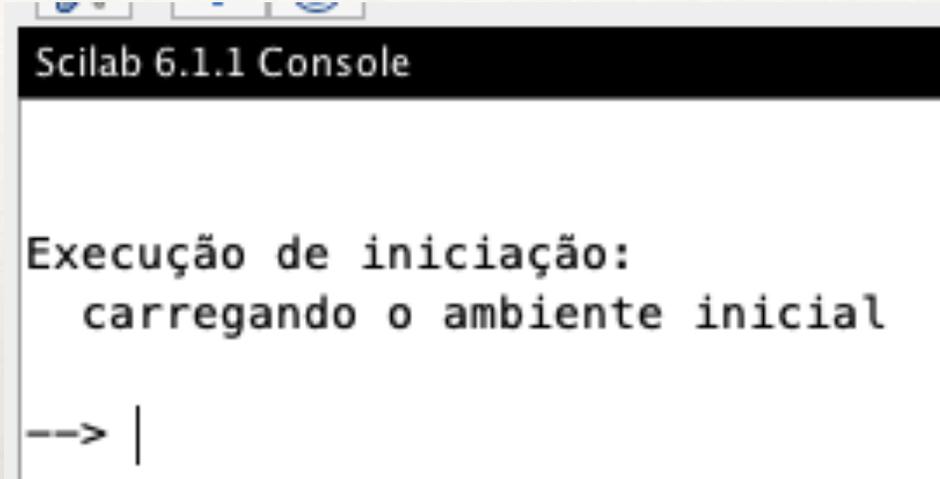
---

No modo interativo, a console oferece o prompt de comando, onde o usuário pode escrever comandos e instruções;

Comandos e instruções recebem respostas após o seu devido processamento. Ex:

**pwd** → indica a pasta atual

**clc** → limpa a console



```
Scilab 6.1.1 Console  
  
Execução de iniciação:  
  carregando o ambiente inicial  
  
--> |
```

---

# Execução de comandos

---

No modo interativo, instrução por instrução, digitada pelo usuário;

Utiliza-se o sinal “;” ao final de cada instrução;

Em *scripts* gravados com a extensão “.sce” contendo diversas instruções;

Os comandos têm sintaxes específicas, e os *scripts* possuem palavras chaves para controle de fluxo, estruturas de repetição etc.

Os *scripts* podem ser editados no SciPad Editor;

---

# Variáveis

---

Armazenam conteúdo para processamento posterior

Podemos armazenar números (reais, inteiros ou complexos) e cadeias de caracteres (strings);

Têm nomes definidos pelo usuário seguindo regras específicas

Maiúsculas e minúsculas são diferentes (“esforco” ≠ “Esforco”);

Tamanho máximo de 24 caracteres;

Não pode começar com números, nem conter espaços ou pontuações;

Existem nomes reservados pelo Scilab (use comando **who**). Ex: **%pi** e **%e**

Atribuição feita pelo operador “=”

Cadeias de caracteres devem ser definidas entre aspas;

Matrizes devem ter elementos declarados entre colchetes ( “[ ]” ), com elementos da mesma linha separados por vírgulas e espaços, e linhas separadas por “ ; ”

# Operadores

## Aritméticos

Operador	Descrição	Ordem de execução
<b>^</b> ou <b>**</b>	Potenciação	Da direita para a esquerda
<b>/</b>	Divisão	Da esquerda para a direita
<b>*</b>	Multiplicação	Da esquerda para a direita
<b>+</b>	Adição	Da esquerda para a direita
<b>-</b>	Subtração	Da esquerda para a direita

Tabela 2.1 - Operadores aritméticos em ordem de prioridade<sup>(6)</sup>.

## Relacionais

Operador	Descrição
<b>&gt;</b>	Maior que
<b>&gt;=</b>	Maior ou igual a
<b>&lt;</b>	Menor que
<b>&lt;=</b>	Menor ou igual a
<b>==</b>	Exatamente igual
<b>&lt;&gt;</b> ou <b>~=</b>	Diferente de

Tabela 2.2 - Operadores relacionais.

## Lógicos:

Operador	Descrição
<b>&amp;</b>	Conjunção
<b> </b>	Disjunção
<b>~</b>	Negação

Tabela 2.3 - Operadores lógicos.

---

# Prioridade dos Operadores

---

1: ~ ==

2: > < ==< >= <> ou ~=

3: ^

4: \* / *Obs. Quando sobre elementos de matrizes, usar um "." antes.*

5: + -

6: &

7: |

Considerar sempre da esquerda para a direita;

Parênteses podem ser usados para mudar prioridades.

---

# Um exemplo de operação

---

—> a = 1;

—> b = 3;

—> c = -10;

—> delta = b\*b - 4\*a\*c;

Qual o valor de delta ?

—> x1 = ( -b + sqrt(delta) ) / 2\*a

Qual o valor de x1 ?

—> x2 = ( -b - sqrt(delta) ) / 2\*a

Qual o valor de x2 ?

—> printf( "%f" , x2 );

O que aconteceu?

---

# Programação - Conceitos

---

O dispositivo computacional executa comandos e instruções sequencialmente, na mesma ordem em que os mesmos são organizados;

O conjunto de comandos e instruções é chamado de programa (no SciLab, e em alguns outros ambientes, chama-se de “*script*”);

Além da declaração de variáveis e comandos, são necessários outros dois tipos de instrução:

- Estruturas de decisão;

- Estruturas de seleção;

- Estruturas de repetição

---

# Estruturas de Decisão

---

se (condição) então

    Instrução x;

    Instrução y;

    . . .

elseif (condição 2) então

    Instrução k;

    Instrução s;

    . . .

Senão     [opcional]

    Instrução z;

    Instrução w;

    . . .

Fim\_se

if(x < y) then

    x=y\*3

    y=y+1

elseif(x == y) then

    x=x+1

    y=y+1

else

    y=x\*3

    x=x+1

end

---

# Estruturas de Decisão II

---

se (condição1) então

    Instrução x;

    . . .

Senão Se (condição2) então [opcional]

    Instrução y;

    . . .

Senão Se (condição3) então [opcional]

    Instrução z;

    . . .

Senão

    Instrução w;

    . . .

Fim\_se

---

# Estruturas de Seleção

---

Selecione (expressão)

caso1

Instrução x;

. . .

caso2

Instrução y;

. . .

caso3

Instrução z;

. . .

caso contrário

Instrução w;

. . .

Fim\_selecione

---

# Estruturas de Repetição

---

Enquanto (condição) faça

    instrução x;

    instrução y;

    . . .

Fim\_enquanto

---

# Estruturas de Repetição II

---

```
Para <varCont>=<inicio>, [Passo], <Fim>
```

```
    instrução x;
```

```
    instrução y;
```

```
    . . .
```

```
Fim_para
```

---

# Programação - Exemplos(1)

---

- 1) Calcule a soma dos 50 primeiros números naturais;
- 2) No exemplo anterior, calcule a última parcela para uma soma de 10.000;
- 3) Dado um vetor de 8 elementos, identifique o maior e o menor elemento, e o produto entre os dois valores.

---

# Exemplo 01:

---

```
soma=0
```

```
numero=1
```

```
while( numero <= 50 )
```

```
    soma = soma + numero
```

```
    numero = numero + 1
```

```
end
```

```
disp(soma)
```

---

# Exemplo 02:

---

```
soma=0
```

```
numero=1
```

```
while( soma <= 10000 )
```

```
    soma = soma + numero
```

```
    numero = numero + 1
```

```
end
```

```
disp(numero)
```

---

# Exemplo 03: (parte I)

---

```
relacao = [100,15,120,2500,5000,2,30,-5]
```

```
i=1
```

```
menor = relacao(i)
```

```
maior = relacao(i)
```

```
while( i <= 8 )
```

```
    if( menor > relacao(i) )
```

```
        menor = relacao(i)
```

```
    end
```

```
    if( maior < relacao(i) )
```

```
        maior = relacao(i)
```

```
    end
```

```
    i = i + 1
```

```
end
```

---

# Exemplo 03: (parte II)

---

```
disp( "Maior: " )
```

```
disp( maior )
```

```
disp( "Menor: ")
```

```
disp( menor )
```

```
disp( "Produto do menor pelo maior:")
```

```
disp( maior * menor )
```

---

# Programação - Exemplos(2)

---

- 1) Ordene um vetor usando o método “Bubble Sort”;
- 3) Dado um número, determine se o mesmo é primo.

```
relacao=[20,19,18,17,16,15,14,13];
troca=1;
while(troca==1)
    i=1;
    troca=0;
    while(i<8)
        if(relacao(i)>relacao(i+1))
            temp=relacao(i);
            relacao(i)=relacao(i+1);
            relacao(i+1)=temp;
            troca=1;
        end
        i=i+1;
    end
end
```

```
i=1;
while(i<9)
    disp(relacao(i));
    i=i+1;
end
```

## Exemplo 2.1:

```
vetor = [10,9,8,7,6,5,4,3,2,1]
disp( "Vetor antes da ordenação:")
disp(vetor)
```

# Exemplo 02:

```
i=1
j=2
ordenando=1
troquei = 0
while (ordenando)
    if( vetor(i) > vetor(j) )
        temporario = vetor(i)
        vetor(i) = vetor(j)
        vetor(j) = temporario
        troquei=1
    end
    i=j
    j=j+1
    if( j==11 )
        if( troquei)
            troquei = 0
            i=1
            j=2
        else
            ordenando=0
        end
    end
end
end
disp( "Vetor depois da ordenação:")
disp( vetor )
```