

# Introdução ao Scilab



Prof. Jeferson Souza, MSc. (thejefecomp)  
[thejefecomp@neartword.com](mailto:thejefecomp@neartword.com)



# O que é o Scilab?

### Definição

O Scilab é um ambiente de computação científica que possibilita o desenvolvimento de soluções para resolver problemas técnicos e/ou científicos. Com o Scilab é possível desenvolver programas (i.e. *software*) com uma linguagem de programação própria, realizar análise numérica e simulações computacionais, bem como gerar gráficos a partir das referidas análises e simulações feitas. Exemplos de ambientes similares incluem o Matlab e o Octave.

## 2 / 49

## O Ambiente do Scilab

## O ambiente do Scilab

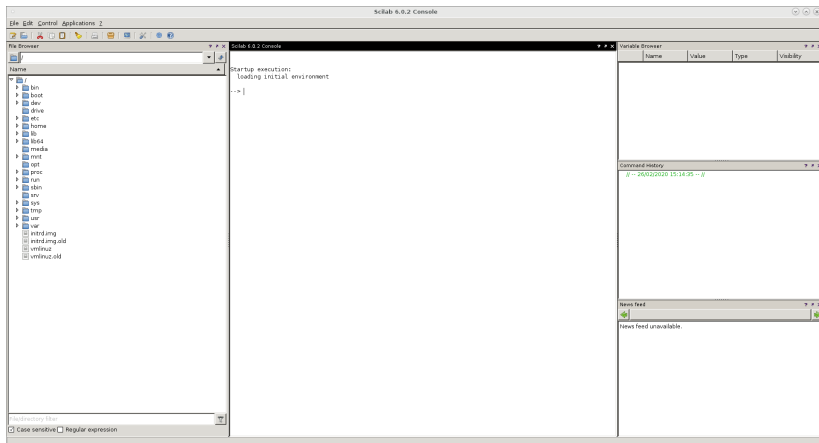
A ferramenta Scilab possui um ambiente de desenvolvimento que possibilita diversas formas de utilização, desde o uso de uma interface gráfica [opção mais comum] até o uso de dois modos console distintos [e.g. presentes na versão 6.0.2 utilizada como base deste material].

# Interface Gráfica de Utilização

## Descrição

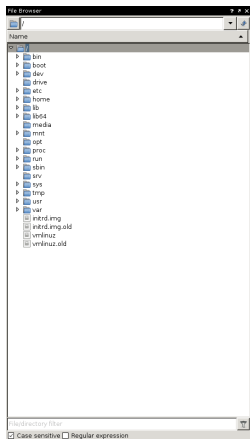
A interface gráfica de utilização [usualmente] inicia-se com cinco componentes distintos agrupados em uma janela única: o navegador de arquivos, o console, o navegador de variáveis, o histórico de comandos, e o painel de notícias.

# Interface Gráfica de Utilização



Interface gráfica de utilização do Scilab.

# Navegador de Arquivos

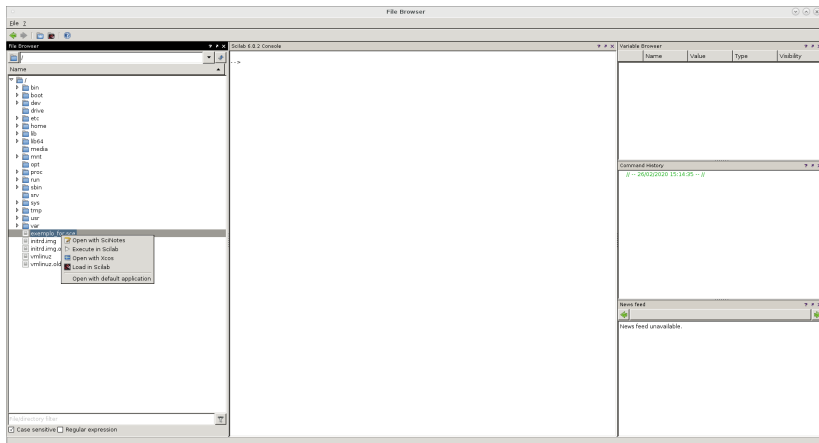


Navegador de Arquivos da Interface Gráfica do Scilab.





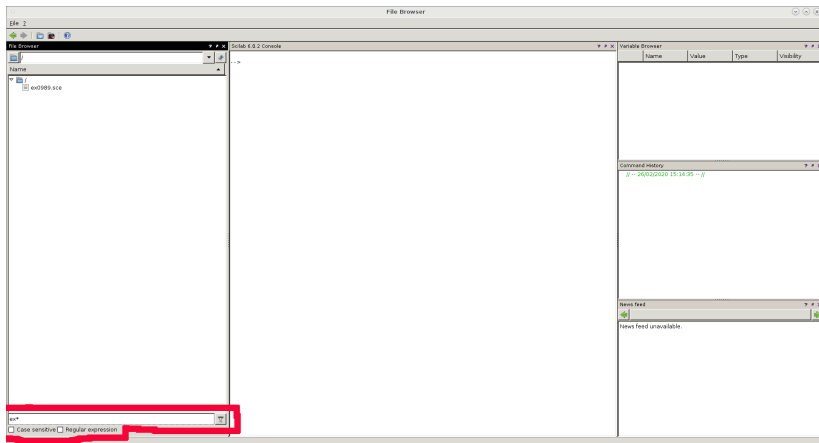
# Navegador de Arquivos



## Seleção de Arquivo no Navegador de Arquivos do Scilab.



# Navegador de Arquivos



## Busca e Filtro de Arquivos no Navegador de Arquivos do Scilab.

## Console

### Descrição

O console é o componente do Scilab responsável pela execução dos programas, a permitir o seu carregamento, sua execução, e a obtenção de resultados.

## Console






Console da Interface Gráfica do Scilab.

## Navegador de Variáveis

### Descrição

O navegador de variáveis permite a manipulação das variáveis já utilizadas por algum programa executado/carregado pelo Scilab, a possibilitar a visualização e edição de seu valor, sua exclusão, e o desenho de gráficos por meio de simples cliques no *mouse*.

# Navegador de Variáveis

Variable Browser <span>?</span> <span>↑</span> <span>×</span>				
	Name	Value	Type	Visibility
	operando1	10	Double	local
	operando2	20	Double	local
	resultado	30	Double	local

Navegador de Variáveis da Interface Gráfica do Scilab.





# Histórico de Comandos



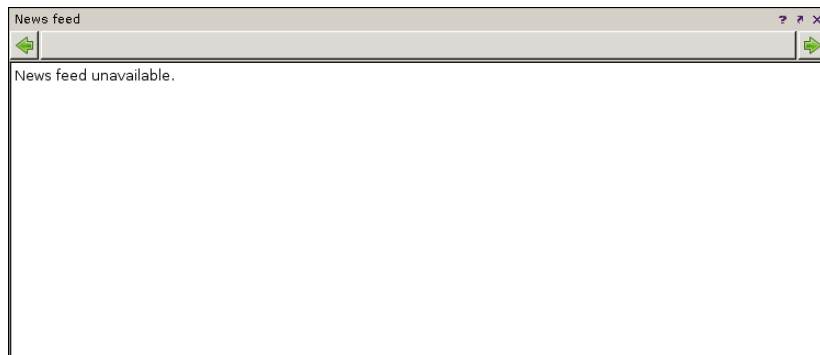
The screenshot shows a window titled "Command History" with a standard macOS-style title bar (red, yellow, green buttons and a close button). The window contains a list of commands executed in the Scilab environment. The first command is a comment: `// -- 26/02/2020 15:14:35 -- //`. The subsequent commands are `operando1 = 10`, `operando2 = 20`, and `resultado = operando1 + operando2`. The window has a scroll bar on the right side.

```
Command History
▼ // -- 26/02/2020 15:14:35 -- //
  operando1 = 10
  operando2 = 20
  resultado = operando1 + operando2
```

Histórico de Comandos da Interface Gráfica do Scilab.



# Painel de Notícias



Painel de Notícias da Interface Gráfica do Scilab.

# A Linguagem de Programação Scilab

## A programar com Scilab

O Scilab possui uma linguagem de programação de alto nível que possibilita o desenvolvimento de algoritmos, de forma muito similar ao processo de desenvolvimento com outras linguagem de programação tal como o C.



# Convenções Adotadas Para o Desenvolvimento de Programas (Continuação)

4. Começar o nome de variáveis SEMPRE com uma letra minúscula, a utilizar o *CamelCase* na sua nomeação.  
Exemplo: valorDaOperacao;
5. Escrever o nome de constantes SEMPRE com todas as letras maiúsculas, com as diferentes palavras separadas pelo caractere underscore (\_). Exemplo: TAMANHO\_LIMITE;
6. Utilizar nomes de funções que sejam sugestivos, consoante seu contexto e utilização; Exemplo: soma ao invés de aa;

# Convenções Adotadas Para o Desenvolvimento de Programas (Continuação)

7. Começar o nome de funções SEMPRE com uma letra minúscula, a utilizar o *CamelCase* na sua nomeação.  
Exemplo: somaInteiro;
8. Declarar o corpo das funções sempre no início do programa;
9. Descrever sucintamente em um comentário [e.g. uma linha] o funcionamento de partes mais complexas do programa.
10. Não copiar a solução do coleguinha sem a perceber por completo. Neste caso, a cópia deixa de fazer sentido :-D.

# A Linguagem de Programação Scilab

## Scilab usa cálculo matricial

Apesar de ser imperceptível, todos os cálculos realizados no Scilab são matriciais. Por isso a facilidade de trabalhar com vetores e matrizes. Números singulares são representados por meio de matrizes de dimensão  $1 \times 1$ .



# A Linguagem de Programação Scilab

## Scilab usa cálculo matricial

Apesar de ser imperceptível, todos os cálculos realizados no Scilab são matriciais. Por isso a facilidade de trabalhar com vetores e matrizes. Números singulares são representados por meio de matrizes de dimensão  $1 \times 1$ .

## Excelente para cálculo numérico

O fato de ter suporte nativo a vetores e matrizes faz do Scilab um ambiente muito interessante para o desenvolvimento de projetos com atividades intensivas de cálculo numérico.

## Variáveis

## Nome de variável

O nome de uma variável pode começar com uma letra ou com um dos seguintes caracteres especiais: %, #, !, \$, ?, e o underscore (\_). Os demais caracteres podem ser alfanuméricos ou um dos caracteres especiais #, !, \$, ?, \_.

## Tamanho e distinção entre maiúsculas e minúsculas

Nomes de variáveis podem ter qualquer tamanho mas somente **os primeiros 24 caracteres** são considerados. Os nomes de variável são **case-sensitive**, o que significa que o Scilab distingue variáveis com letras maiúsculas e minúsculas.

## Variáveis

## Variáveis iniciadas com o caracter especial %

Variáveis iniciadas com caracter especial % usualmente representam constantes internas do Scilab ou funções que sobrecarregam operadores.

## Exemplos:

%pi \\ representa o valor do  $\pi = 3.1415...$

`%inf` \representa o infinito  $[\infty]$ .

# Variáveis

## Representação da entrada padrão [teclado] e da saída padrão [console]

No Scilab o acesso a entrada padrão, o qual permite ao utilizador fornecer dados por meio do teclado, é representado por `%io(1)`; enquanto que o acesso a saída padrão, o qual permite a exibição de mensagens na janela do console, é representado por `%io(2)`.

# Variáveis

## Declaração de variáveis

No Scilab a declaração de uma variável acontece no momento de sua primeira atribuição de valor. Não é necessário [tal como no C] declarar o tipo de variável antes de sua utilização.

## Exemplos

→ *nota* = 8.5

→ *mensagem* = “Porqueosolparecedistante?”

# Variáveis

## Declaração de variáveis

No Scilab a declaração de uma variável acontece no momento de sua primeira atribuição de valor. Não é necessário [tal como no C] declarar o tipo de variável antes de sua utilização.

## Exemplos

→ *nota* = 8.5

→ *mensagem* = “Porque o sol parece distante?”

## Pegadinha... :-D

“Por que o sol parece distante?” [Ficou mais fácil ler a frase? :-D]

# O uso do ponto e vírgula [;]

## O uso do ponto e vírgula [;]

O uso do ponto e vírgula [;] na terminação de operações indica ao Scilab que o resultado da operação não deve ser exibido. Entretanto, o ponto e vírgula [;] também pode ser utilizado na representação de matrizes, a indicar elementos de uma nova linha.

# Meu Primeiro Programa em Scilab

## Console

Para escrever nosso primeiro programa em Scilab, abra a interface gráfica, vá ao console, e digite o seguinte comando:

→ *disp("Olá Mundo!")*



# Meu Primeiro Programa em Scilab

## No Editor de Texto [e.g. SciNotes]

Para escrever nosso primeiro programa em Scilab por meio de um editor de texto, abra o seu editor de texto favorito [e.g. SciNotes], siga as convenções, e inclua o seguinte comando:

```
disp("Olá Mundo!")
```

Salve o arquivo com a extensão .sce e execute-o no Scilab.

# Operadores Aritméticos

O Scilab possui uma lista de operadores aritméticos para utilização. São eles:

+	Adição
-	Subtração
*	Multiplicação de matriz
.*	Multiplicação de vetor
.*.	Multiplicação de Kronecker
/	Divisão
\	Divisão esquerda de matriz

# Operadores Aritméticos - Continuação

$\./$	Divisão de vetor
$.\backslash$	Divisão esquerda de vetor
$./.$	Divisão de Kronecker
$.\backslash.$	Divisão esquerda de Kronecker
$^$ ou $**$	Exponenciação de matriz
$.^$	Exponenciação de vetor
$'$	Transposição de matriz complexa
$.'$	Transposição de vetor

# Operadores de Comparação

Os operadores de comparação do Scilab são os seguintes:

<	Menor que
>	Maior que
<=	Menor ou igual a
>=	Maior ou igual a
==	Igual a
<> ou ~ =	Diferente de
&&	E Lógico
	OU Lógico

# Estruturas de Controle

O Scilab possui as seguintes estruturas de controle:

<b>break</b>	Força o término de um laço
<b>case</b>	Inicia uma cláusula em um bloco <b>select</b>
<b>elseif</b>	Inicia um bloco condicional alternativo a um bloco <b>if</b>
<b>else</b>	Inicia uma alternativa em ambos os blocos <b>if</b> e <b>select</b>
<b>end</b>	Termina os blocos <b>for</b> , <b>if</b> , <b>select</b> , e <b>while</b>
<b>errcatch</b>	Possibilita o tratamento de erros
<b>for</b>	Inicia um laço usualmente com repetições pré-estabelecidas

# Estruturas de Controle - Continuação

if	Inicia um bloco condicional que pode conter múltiplas alternativas
select	Inicia um bloco de seleção que pode conter múltiplas alternativas
while	Inicia um laço cuja execução é repetida até que uma condição seja satisfeita

# Estrutura Condicional - if

## Estrutura condicional if

```
if <condicao1> then
```

```
    <comandos>
```

```
elseif <condicao2>
```

```
    <comandos>
```

```
elseif <condicao3>
```

```
    <comandos>
```

```
else
```

```
    <comandos>
```

```
end
```

# Estrutura Condicional - if

## Exemplos

```
if x < 10 then
```

```
    disp('O numero é menor do que 10')
```

```
end
```

```
if nota >= 7 then
```

```
    disp('O aluno foi aprovado')
```

```
else
```

```
    disp('O aluno foi para exame')
```

```
end
```



# Estrutura de Seleção - **select**

## Estrutura de seleção **select**

**select** <expressão>

**case** <condição1> **then** <comandos>

**case** <condição2> **then** <comandos>

**case** <condiçãon> **then** <comandos>

**else** <comandos>

**end**

A <expressão> precisa ser igual a <condição> de seleção especificada.

# Estrutura de Seleção - **select**

## Exemplo

```
select opcao  
  case 1 then disp('aluno aprovado')  
  case 2 then disp('aluno reprovado')  
  else disp('opcao inexistente')  
end
```

# Estrutura de Repetição - **for**

## Estrutura de repetição **for**

**for** <variável>=<início>:<fim>

<comandos>

**end**

# Estrutura de Repetição - **for**

## Exemplo

```
soma = 0
```

```
for i=1:5
```

```
    soma = soma + 10
```

```
end
```

# Estrutura de Repetição - **while**

## Estrutura de repetição **while**

**while** <condição>

    <comandos>

**end**

# Estrutura de Repetição - **while**

## Exemplo

```
soma = 0
```

```
while soma < 200
```

```
    soma = soma + 10
```

```
end
```

# Leitura do teclado

A leitura dos valores provenientes do teclado pode ser feito por meio da função *input()*.

## Função *input()*

*input*(<mensagem>, <indicador\_de\_caracter>)

Onde:

<mensagem> indica a mensagem que será inserida para que o utilizador informe a entrada;

<indicador\_de\_caracter> é opcional e indica que a entrada é uma cadeia de caracteres.

# Leitura do teclado - Exemplo

## Exemplos de utilização

→ `nota1 = input('Informe a nota: ')`

→ `nome = input('Informe o nome: ', 's')`



# Escrita no console

A escrita no console pode ser realizada por meio de diferentes funções tais como a função *disp()* e a função *mprintf()*.

## Função *disp()*

*disp*(<argumento1>, <argumento2>, ..., <argumenton>)

Onde <argumenton> representa uma variável ou uma mensagem cujos valores serão exibidos no console.

# Escrita no console

## Função **disp()**

**disp**(<argumento1>, <argumento2>, ..., <argumenton>)

Onde <argumenton> representa uma variável ou uma mensagem cujos valores serão exibidos no console.

## Exemplos de utilização

→ **disp**(nota1)

→ **disp**('O nome eh: ', nome) ⇔ **disp**(nome, 'O nome eh: ') \*

\* Os argumentos são exibidos em ordem inversa, ou seja, da direita para a esquerda.

# Escrita no console

## Função **mprintf()**

*mprintf*(<formato>, <variável1>, <variável2>, <variáveln>)

Onde:

<formato> representa o formato de exibição no console;

<variável1...n> representa a variável cujo valor formatado será exibido no console.

# Escrita no console - Formato

<code>%ni</code>	exibe números inteiros com tamanho n
<code>%n.df</code>	números de ponto flutuante com d casas decimais e tamanho n
<code>%n.de</code>	exibe número em notação exponencial com d casas decimais e tamanho n
<code>%n.dg</code>	exibe o formato mais curto entre os formatos <code>%n.df</code> e <code>%n.de</code>
<code>%ns</code>	exibe uma cadeia de caracteres com tamanho n

# Escrita no console - Exemplos Função **mprintf()**

## Exemplos de utilização da função **mprintf()**

→ **mprintf**('O valor é: %3i', valor)

→ **mprintf**('A nota é: %3.2f', nota)

# Bibliografia



SCILAB ENTERPRISES *"Scilab Online Help"*. 2019.  
Disponível em: [https://help.scilab.org/docs/6.0.2/en\\_US/index.html](https://help.scilab.org/docs/6.0.2/en_US/index.html).  
Acesso em: 03 Jun. 2021.

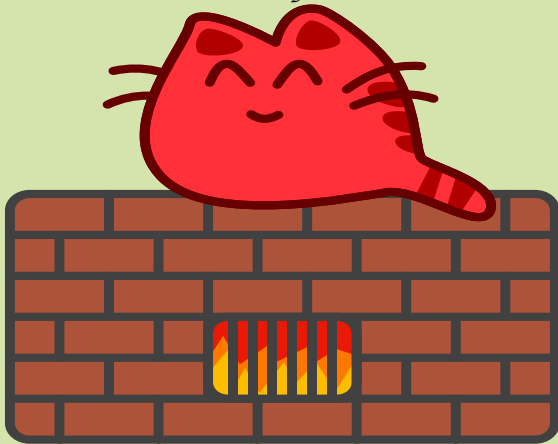


Gomez, C. and Scilab Enterprises. *"Scilab for very beginners"*, 2013.



Rietsch, E. *"An Introduction to Scilab from a Matlab User's Point of View"*. version 2.6-1.0. 2001-2002.

*That's it folks!*



*Thank you for your attention!*