

Introdução à Programação de Computadores para Biologia

Introdução a Unix

UNIX

PREPARAÇÃO

1. Na página da disciplina, baixar os seguintes arquivos:

- Arquivo em formato fasta: **metagenomics_coral.fasta**
- Arquivo em formato fasta: **dmel-subset-gene-r5.45.fasta**
- Arquivo em formato fastq: **chom.txt**
- Arquivo: **prova1.tar.gz**

Link:

<https://tttorres.github.io/introprog/>

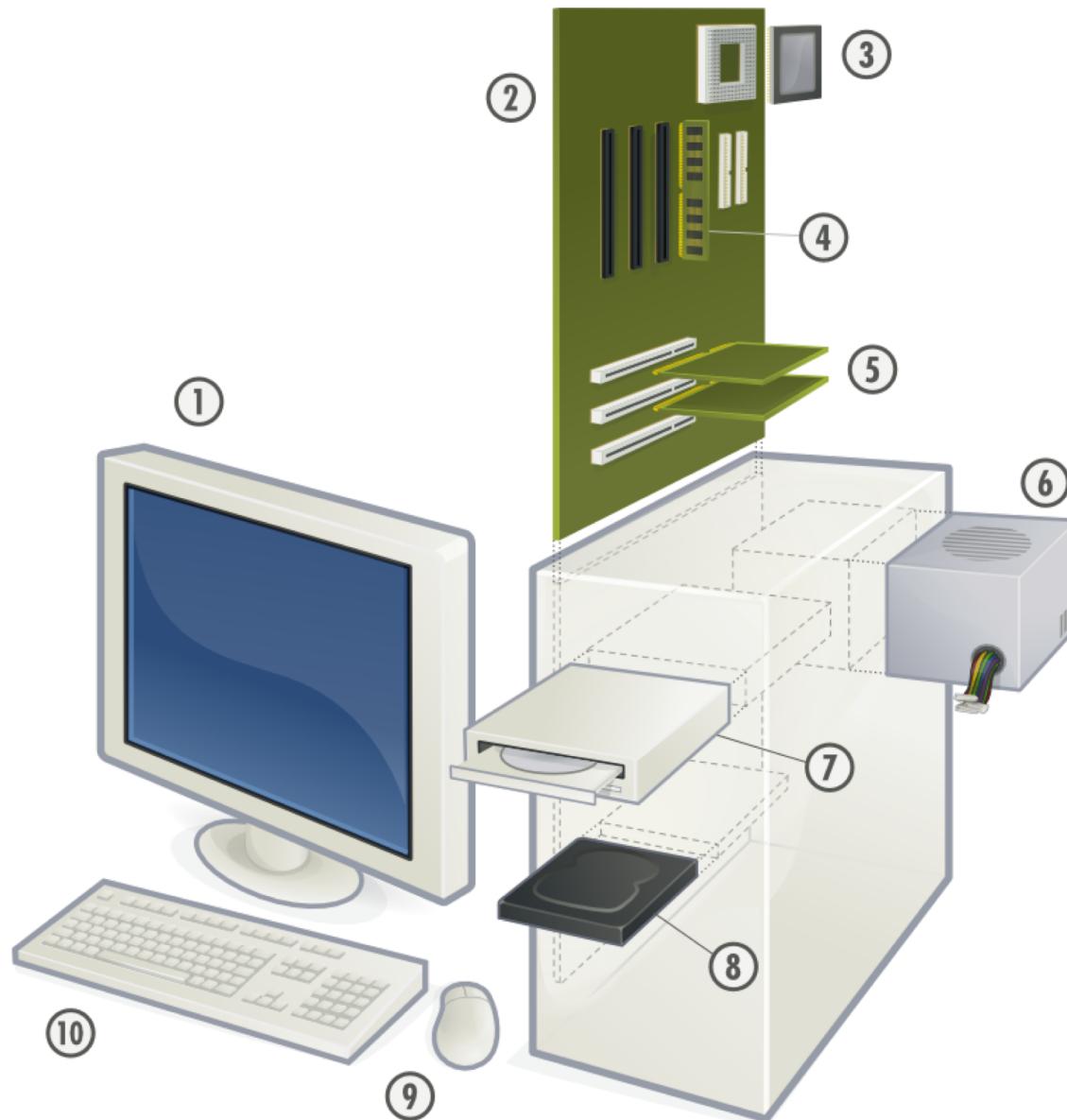
Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

COMPUTADOR

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

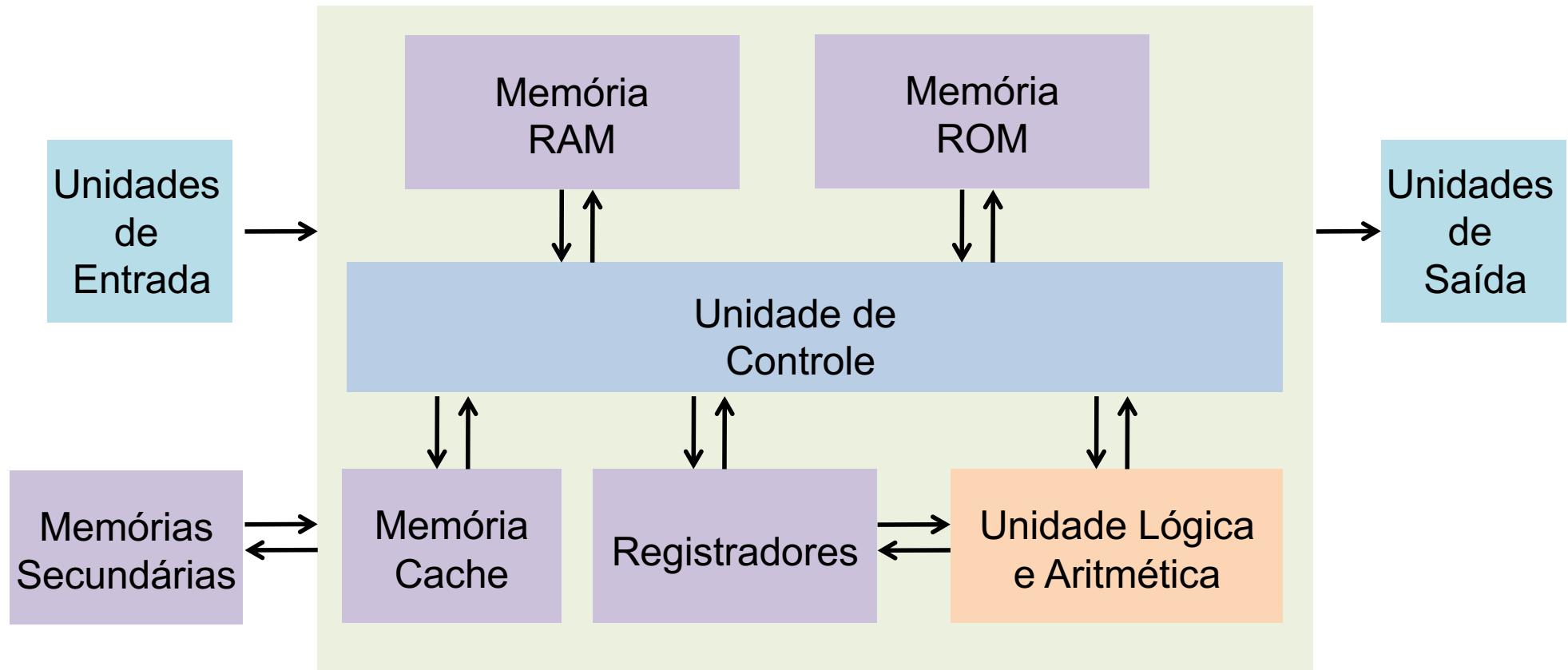


Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

HARDWARE

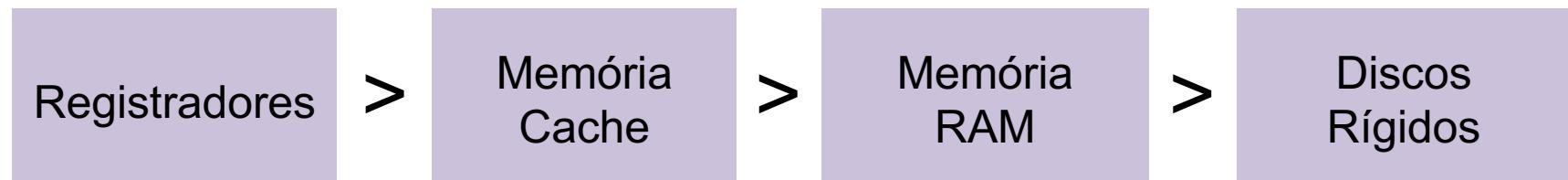
Unidade Central de Processamento



Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

HARDWARE



Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

HARDWARE -> SOFTWARE

HARDWARE = conjunto de interruptores

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

HARDWARE -> SOFTWARE

HARDWARE = conjunto de interruptores

“0” desligado
“1” ligado

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

HARDWARE -> SOFTWARE

HARDWARE = conjunto de interruptores

“0” desligado
“1” ligado

Unidade de informação = *bit*

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

HARDWARE -> SOFTWARE

HARDWARE = conjunto de interruptores

“0” desligado
“1” ligado

1 bit pode representar dois valores 0 e 1

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

HARDWARE -> SOFTWARE

HARDWARE = conjunto de interruptores

2 bits podem representar quatro valores:

| bit1bit2 | valor |
|----------|-------|
| 00 | 0 |
| 01 | 1 |
| 10 | 2 |
| 11 | 3 |

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

HARDWARE -> SOFTWARE

3 bits podem representar oito valores:

| bit1bit2 | valor |
|----------|-------|
| 000 | 0 |
| 001 | 1 |
| 010 | 2 |
| 011 | 3 |
| 100 | 4 |
| 101 | 5 |
| 110 | 6 |
| 111 | 7 |

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

HARDWARE -> SOFTWARE

3 bits podem representar oito valores:

| bit1bit2 | valor | 2^n |
|----------|-------|-------|
| 000 | 0 | |
| 001 | 1 | |
| 010 | 2 | |
| 011 | 3 | |
| 100 | 4 | |
| 101 | 5 | |
| 110 | 6 | |
| 111 | 7 | |

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

HARDWARE -> SOFTWARE

HARDWARE = conjunto de interruptores

“0” desligado
“1” ligado

Unidade de informação = *bit*

Conjunto de 8 *bits* = *byte*
(é, tradicionalmente, o número de bits para representar um caracter)

1 *byte* pode ter até $2^8 = 256$ configurações diferentes

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

HARDWARE -> SOFTWARE

| NOME | MEMÓRIA |
|------------------|--|
| bit | {0,1} |
| byte | 8 bits |
| kilobyte (kbyte) | 2^{10} bytes (1024) |
| megabyte | 2^{20} bytes (pouco mais de 1 milhão de bytes) |
| gigabyte | 2^{30} bytes (pouco mais de 1 bilhão de bytes) |

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

BASE DECIMAL

526

$$526 = 5 * 10^2 + 2 * 10^1 + 6 * 10^0$$

$$526 = 500 + 20 + 6$$

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

BASE DECIMAL

38

$$38 = 3 * 10^1 + 8 * 10^0$$

$$526 = 30 + 8$$

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

BASE DECIMAL

38

$$38 = 3 * 10^1 + 8 * 10^0$$

$$526 = 30 + 8$$

BASE BINÁRIA

100110_2

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

BASE DECIMAL

38

$$38 = 3 * 10^1 + 8 * 10^0$$

$$526 = 30 + 8$$

BASE BINÁRIA

100110_2

$$\begin{aligned}1 * 2^5 + 0 * 2^4 + 0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0 &= \\&= 32 + 0 + 0 + 4 + 2 + 0 = 38\end{aligned}$$

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

CONVERSÃO DE BASE DECIMAL PARA BINÁRIA

EXEMPLO: 126

| | Quociente | Resto | |
|-------------|-----------|-------|--|
| $126 / 2 =$ | 63 | 0 | |
| $63 / 2 =$ | 31 | 1 | |
| $31 / 2 =$ | 15 | 1 | |
| $15 / 2 =$ | 7 | 1 | |
| $7 / 2 =$ | 3 | 1 | |
| $3 / 2 =$ | 1 | 1 | |
| $1 / 2 =$ | | 1 | |

The diagram illustrates the division process for converting the decimal number 126 to binary. A vertical arrow points upwards from the remainders (0, 1, 1, 1, 1, 1, 1) towards the top of the slide, indicating the direction of reading the binary digits.

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

CONVERSÃO DE BASE DECIMAL PARA BINÁRIA

EXEMPLO: 1111110_2

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

O PROGRAMA MAIS ESCRITO NO MUNDO

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793-1



O PROGRAMA MAIS ESCRITO NO MUNDO

Imprimir “Hello, world!” na tela

```
01001000 01100101 01101100 01101100 01101111 00101100  
00100000 01010111 01101111 01110010 01101100 01100100  
00100001
```

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

| | |
|------------------|---|
| 00000000 | 7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00 00 00 00 00 |
| .ELF..... | 00000010 02 00 03 00 01 00 00 00 00 80 80 04 08 34 00 00 00 00 |
|4.. | 00000020 c8 00 00 00 00 00 00 00 00 34 00 20 00 02 00 28 00 |
|4.(. | 00000030 04 00 03 00 01 00 00 00 00 00 00 00 00 00 00 00 80 04 08 |
| | 00000040 00 80 04 08 9d 00 00 00 00 9d 00 00 00 00 05 00 00 00 00 |
| | 00000050 00 10 00 00 01 00 00 00 00 a0 00 00 00 00 a0 90 04 08 |
| | 00000060 a0 90 04 08 0e 00 00 00 00 0e 00 00 00 00 06 00 00 00 00 |
| | 00000070 00 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
| | 00000080 ba 0e 00 00 00 b9 a0 90 04 08 bb 01 00 00 00 b8 |
| | 00000090 04 00 00 00 cd 80 b8 01 00 00 00 00 cd 80 00 00 00 |
| | 000000a0 48 65 6c 6c 6f 2c 20 77 6f 72 6c 64 21 0a 00 2e |
| Hello, world!... | 000000b0 73 68 73 74 72 74 61 62 00 2e 74 65 78 74 00 2e |
| shstrtab..text.. | 000000c0 64 61 74 61 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
| data..... | 000000d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
| | |
| * | |
| 000000f0 | 0b 00 00 00 01 00 00 00 06 00 00 00 80 80 04 08 |
| | 00000100 80 00 00 00 1d 00 00 00 00 00 00 00 00 00 00 00 00 00 |
| | 00000110 10 00 00 00 00 00 00 00 00 11 00 00 00 01 00 00 00 |
| | 00000120 03 00 00 00 a0 90 04 08 a0 00 00 00 0e 00 00 00 |
| | 00000130 00 00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00 00 |
| | 00000140 01 00 00 00 03 00 00 00 00 00 00 00 00 00 00 00 00 00 |
| | 00000150 ae 00 00 00 17 00 00 00 00 00 00 00 00 00 00 00 00 00 |
| | 00000160 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
| | |

Linux 32bit

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

```
section .text
    global _start          ;must be declared for linker
(_ld)_start:           ;tell linker entry point

    mov edx,len ;message length
    mov ecx,msg ;message to write
    mov ebx,1   ;file descriptor (stdout)
    mov eax,4   ;system call number (sys_write)
    int 0x80    ;call kernel

    mov eax,1   ;system call number (sys_exit)
    int 0x80    ;call kernel

section .data
msg db  'Hello, world!',0xa ;our dear string
len equ $ - msg            ;length of our dear string
```

Linux Assembler

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

```
#include <stdio.h>
int main() {
    printf("Hello World!\n");
return 0;
}
```

C

```
10 PRINT "Hello world!"
20 END
```

BASIC

```
public class HelloWorld
{
    public static void main(string[] args)
    {
        System.out.println("Hello world!");
    }
}
```

Java

```
program HelloWorld;
begin
    writeln('Hello World');
end.
```

Pascal

```
#!/usr/bin/perl
print "Hello World!\n";
```

Perl

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

SCRIPTS vs PROGRAMAS

```
#include <stdio.h>
int main() {
    printf("Hello World!\n");
return 0;
}
```

C

```
#!/usr/bin/perl
print "Hello World!\n";
```

PERL

Introdução à Programação de Computadores para Biologia

BIO0454 / BIO5793

PRIMEIRO PROGRAMA (OU SCRIPT)

```
#!/usr/bin/perl
my ($E, $P, $M, $nota_final);

($E, $P, $M) = @ARGV;
$nota_final = ((2*$E)+(3*$P)+(5*$M))/10;
if ($nota_final >= 5) {
    print "Aluno aprovado\n";
} else {
    print "Aluno reprovado\n";
}

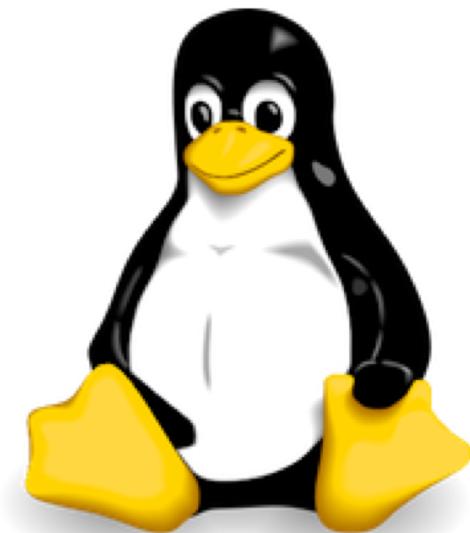
# conversao de nota para conceitos, apenas para a Pos-graduacao

if ($nota_final < 5) {
    print "R, Reprovado, sem direito a credito\n";
} elsif ($nota_final <= 7.0) {

    print "C, Regular, com direito a credito\n";
} elsif ($nota_final <= 8.5) {
    print "B, Bom, com direito a credito\n";
} else {
    print "A, Excelente, com direito a credito\n";
}

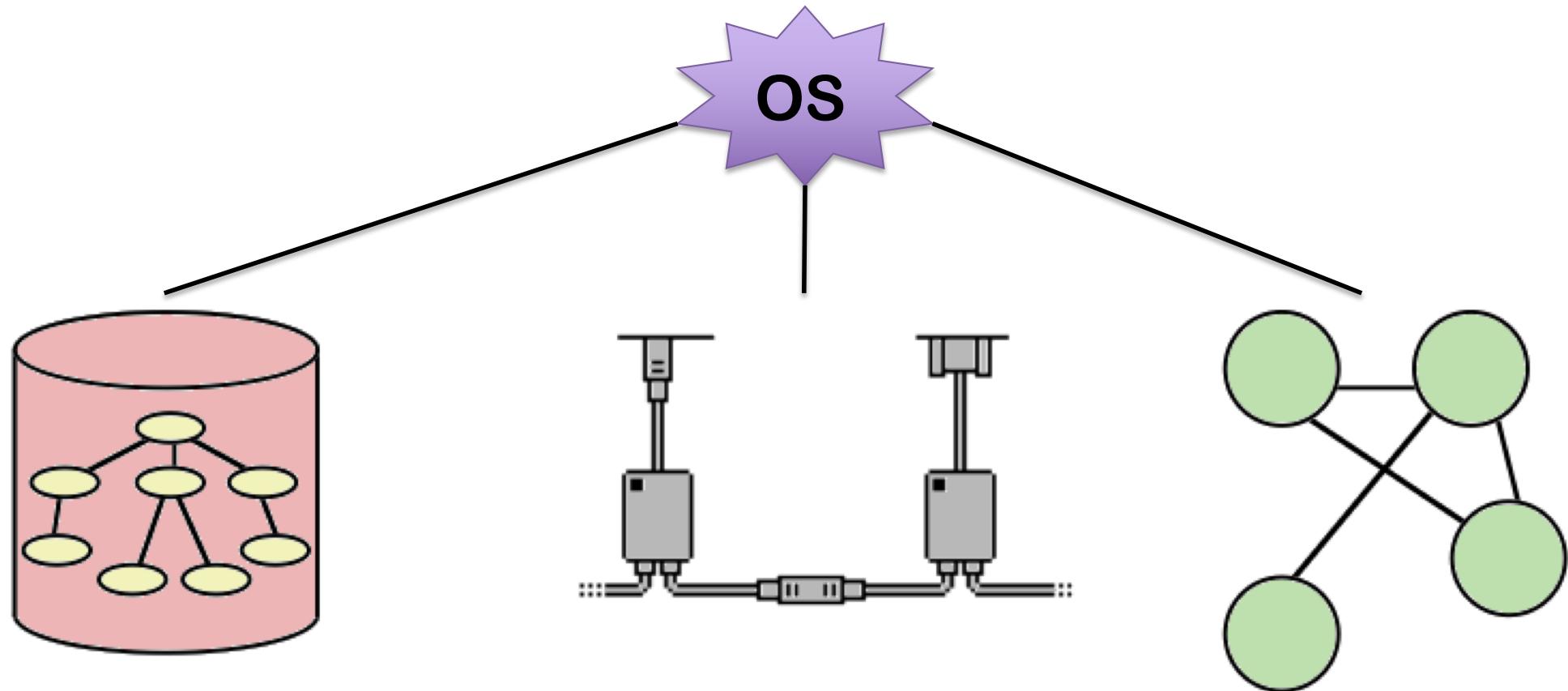
exit;
```

SISTEMA OPERACIONAL



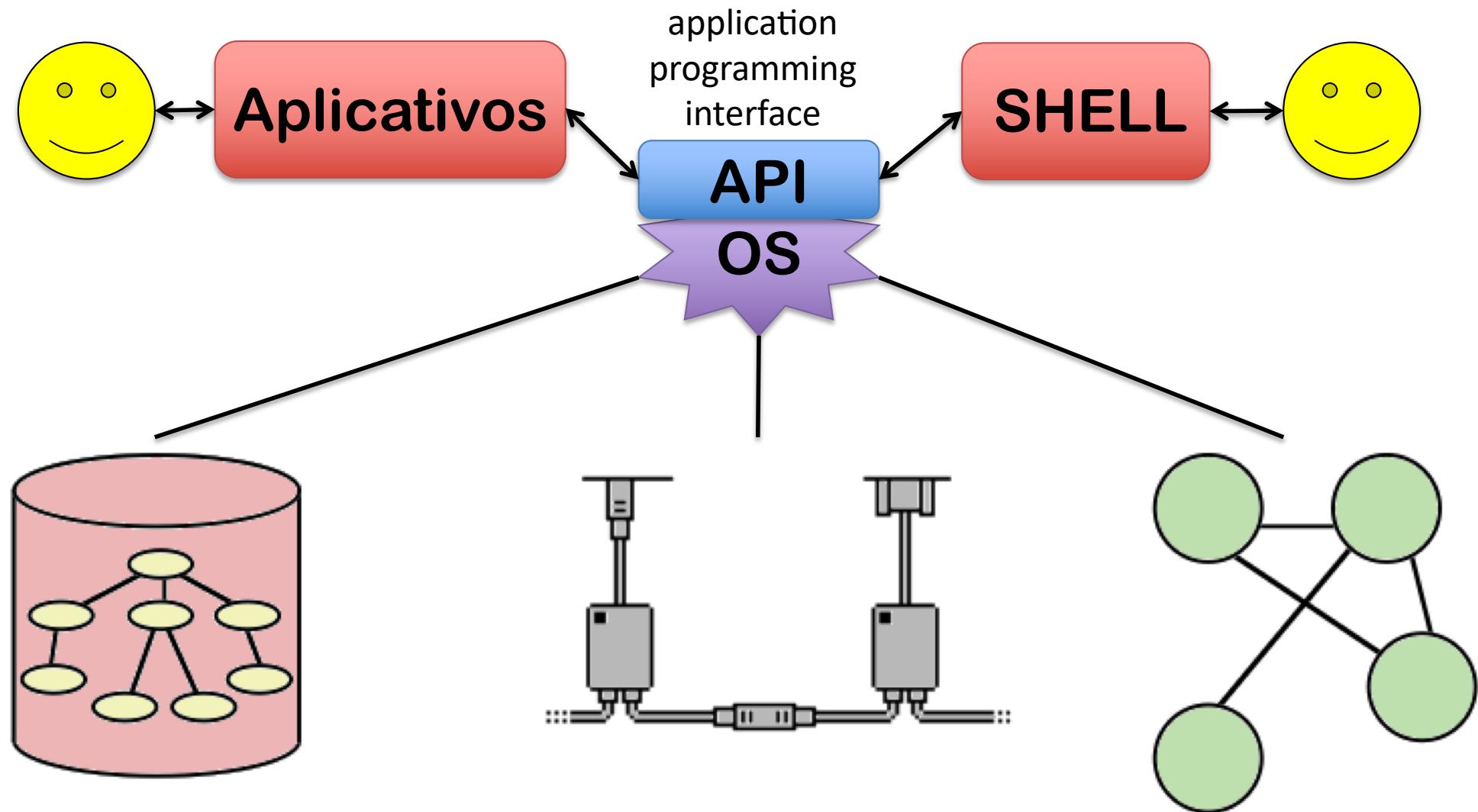
SISTEMA OPERACIONAL

GERENCIADOR DE TAREFAS

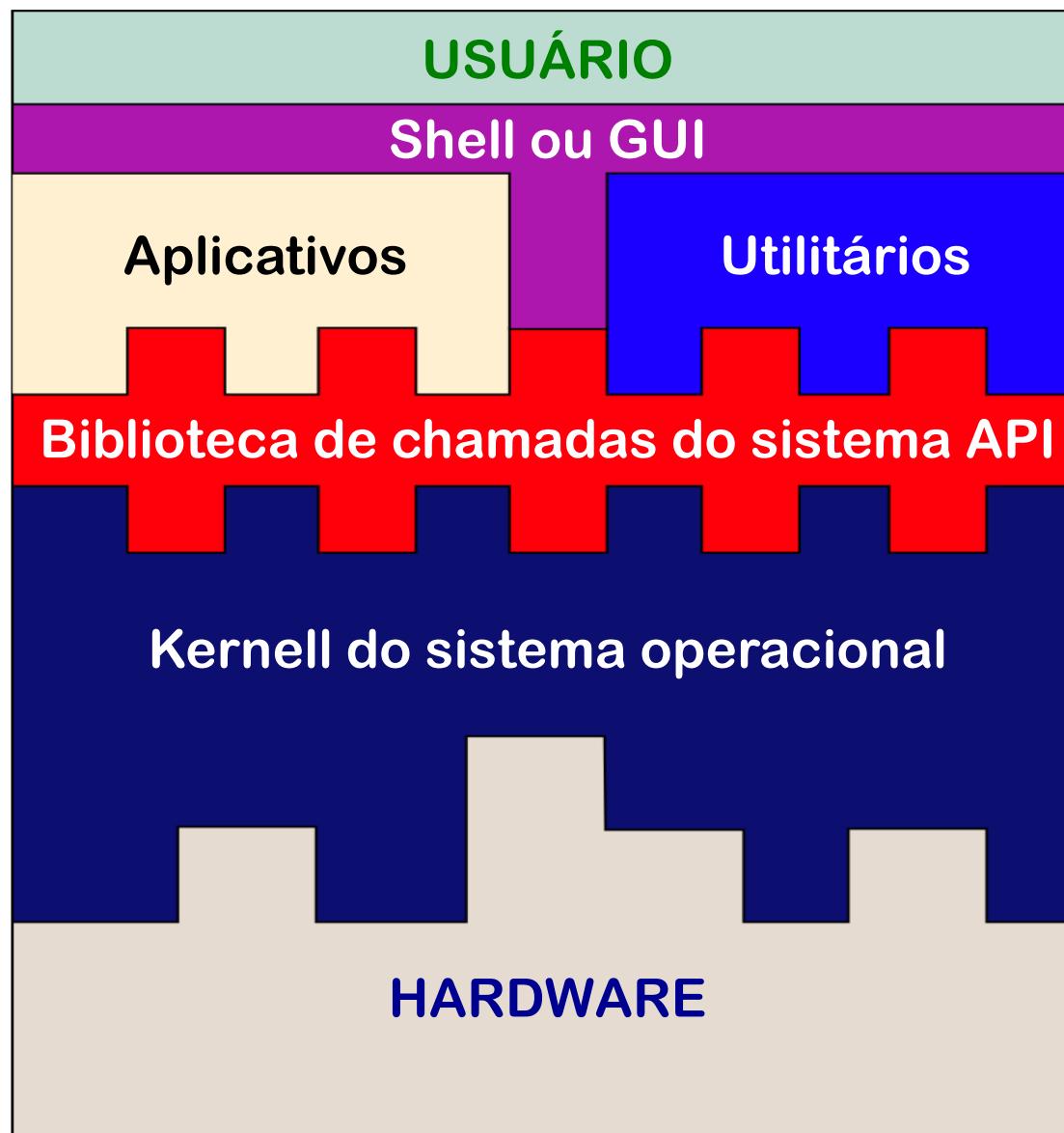


SISTEMA OPERACIONAL

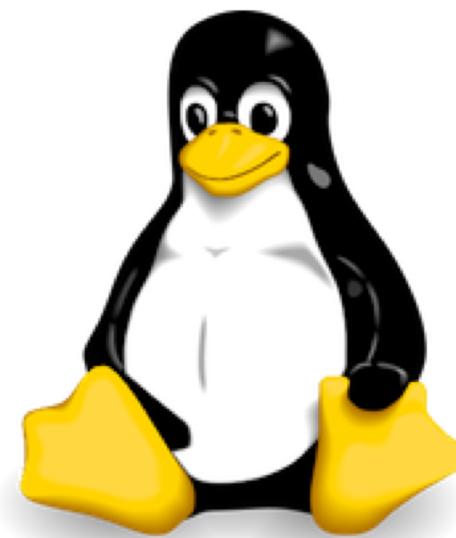
GERENCIADOR DE TAREFAS



SISTEMA OPERACIONAL

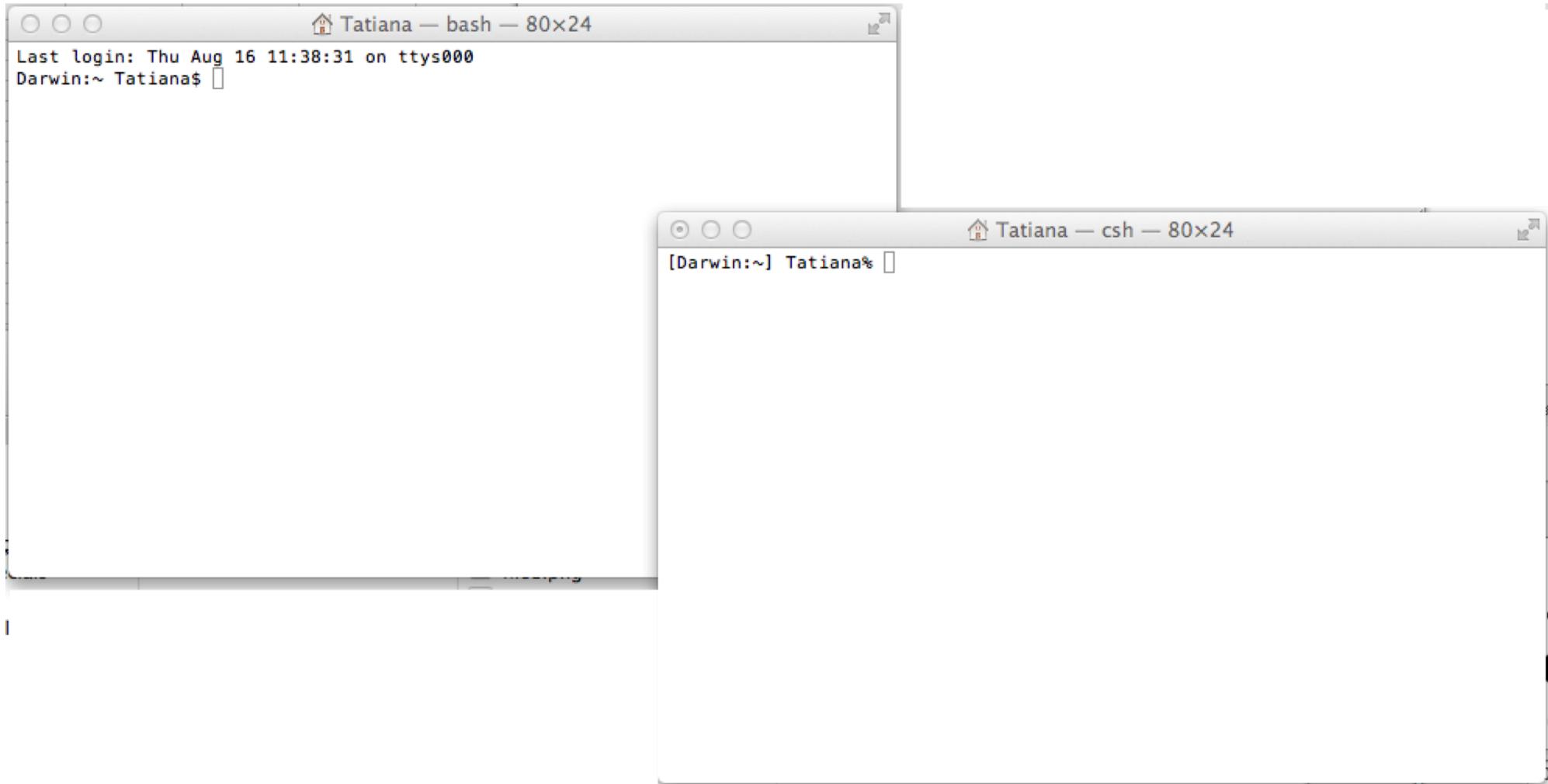


SISTEMA OPERACIONAL



SISTEMA OPERACIONAL

SHELL



UNIX

COMANDOS MAIS UTILIZADOS

1. Comando ls: listando arquivos e diretórios:

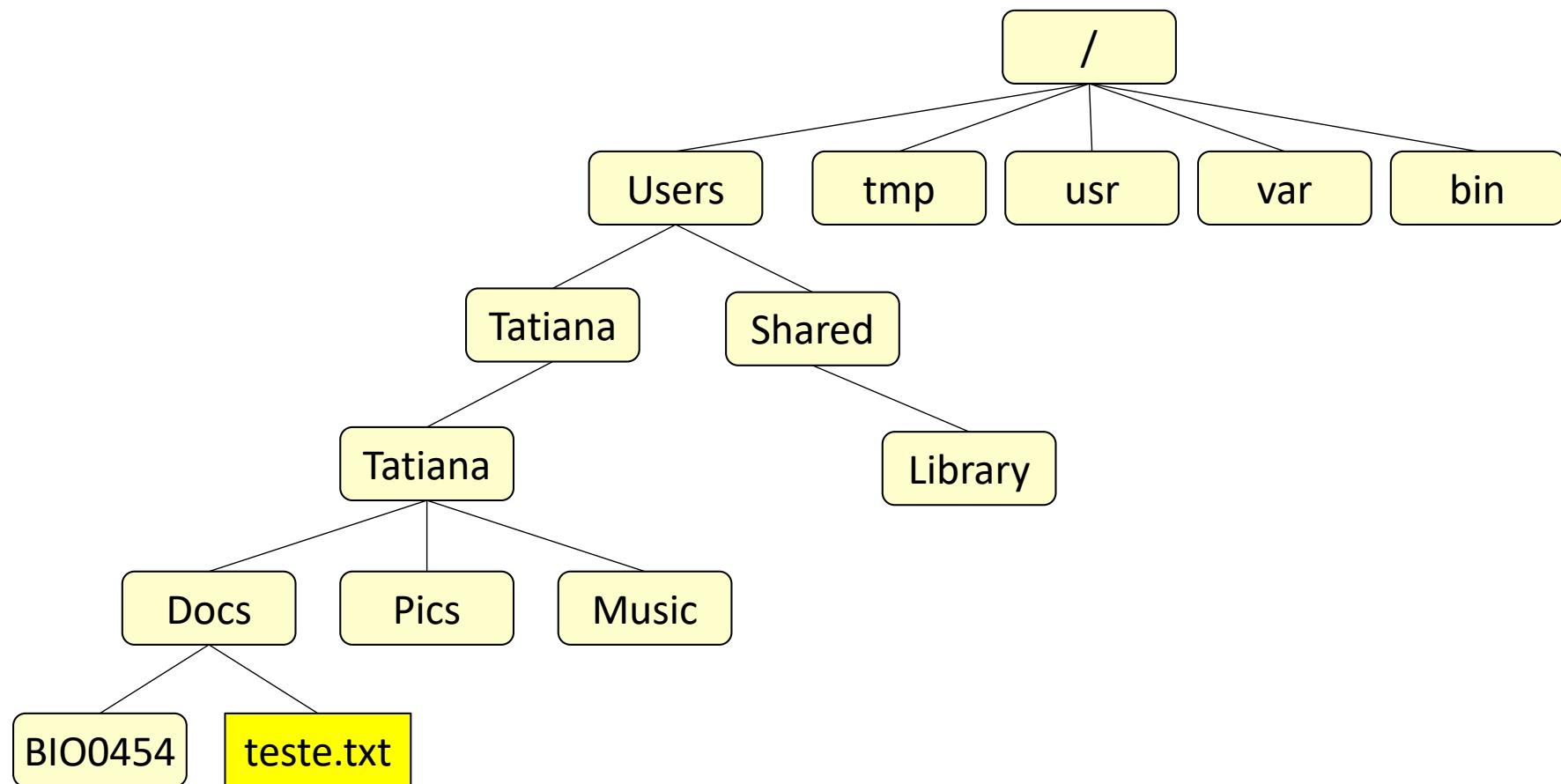
```
ls      #list  
ls -a  
ls -l  
ls -la
```

OBS: man #manual

UNIX

COMANDOS MAIS UTILIZADOS

2. “Navegando” em diretórios:



UNIX

COMANDOS MAIS UTILIZADOS

A. Verificar o diretório atual

B. Ir para a raiz

C. Voltar para home

UNIX

COMANDOS MAIS UTILIZADOS

A. Verificar o diretório atual

```
pwd
```

B. Ir para a raiz

C. Voltar para home

UNIX

COMANDOS MAIS UTILIZADOS

A. Verificar o diretório atual

```
pwd
```

B. Ir para a raiz

```
cd /
```

C. Voltar para home

UNIX

COMANDOS MAIS UTILIZADOS

A. Verificar o diretório atual

```
pwd
```

B. Ir para a raiz

```
cd /
```

C. Voltar para home

```
cd ~
```

UNIX

COMANDOS MAIS UTILIZADOS

3. Criando e alterando arquivos e diretórios

- `mkdir` # make directory
- `cp` # copy
- `mv` # move
- `rm` # remove; MUITA CALMA!!!

UNIX

COMANDOS MAIS UTILIZADOS

A. Criar a pasta myseq no home

B. Mover o arquivo dmel-gene.fasta para myseq

C. Ir para a pasta myseq

UNIX

COMANDOS MAIS UTILIZADOS

A. Criar a pasta myseq no home

```
mkdir myseq
```

B. Mover o arquivo dmel-gene.fasta para myseq

C. Ir para a pasta myseq

UNIX

COMANDOS MAIS UTILIZADOS

A. Criar a pasta myseq no home

```
mkdir myseq
```

B. Mover o arquivo dmel-gene.fasta para myseq

```
mv dmel-gene.fasta ~/myseq/
```

C. Ir para a pasta myseq

UNIX

COMANDOS MAIS UTILIZADOS

A. Criar a pasta myseq no home

```
mkdir myseq
```

B. Mover o arquivo dmel-gene.fasta para myseq

```
mv dmel-gene.fasta ~/myseq/
```

C. Ir para a pasta myseq

```
cd ~/myseq/
```

UNIX

COMANDOS MAIS UTILIZADOS

D. Copiar o arquivo com novo nome: nome.fasta

E. Apagar o arquivo nome.fasta

UNIX

COMANDOS MAIS UTILIZADOS

D. Copiar o arquivo com novo nome: novo.fasta

```
cp dmel-gene.fasta novo.fasta
```

E. Apagar o arquivo novo.fasta

UNIX

COMANDOS MAIS UTILIZADOS

D. Copiar o arquivo com novo nome: novo.fasta

```
cp dmel-gene.fasta novo.fasta
```

E. Apagar o arquivo novo.fasta

```
rm novo.fasta
```

UNIX

COMANDOS MAIS UTILIZADOS

4. Visualização de arquivos (e informações)

- head
- tail
- wc #word count
- grep #globally search a regular expression and print

UNIX

CONTINUAÇÃO

Visualização de arquivos (e informações)

- head

head –número de linhas input

```
head -10 meta_coral.fasta
```

UNIX

CONTINUAÇÃO

Visualização de arquivos (e informações)

- tail

tail -número de linhas input

```
tail -10 meta_coral.fasta
```

UNIX

CONTINUAÇÃO

Visualização de arquivos (e informações)

- md5 / md5sum

md5 input

```
md5 meta_coral.fasta
```

UNIX

CONTINUAÇÃO

Visualização de arquivos (e informações)

- `wc #lines, words and bytes`

`wc input`

```
wc meta_coral.fasta
```

UNIX

COMANDOS MAIS UTILIZADOS

A. Visualizar as primeiras linhas de um arquivo

B. Ver as 20 primeiras linhas do arquivo

C. Visualizar as últimas linhas de um arquivo

UNIX

COMANDOS MAIS UTILIZADOS

A. Visualizar as primeiras linhas de um arquivo

```
head dmel-gene.fasta
```

B. Ver as 20 primeiras linhas do arquivo

C. Visualizar as últimas linhas de um arquivo

UNIX

COMANDOS MAIS UTILIZADOS

A. Visualizar as primeiras linhas de um arquivo

```
head dmel-gene.fasta
```

B. Ver as 20 primeiras linhas do arquivo

```
head -20 dmel-gene.fasta
```

C. Visualizar as últimas linhas de um arquivo

UNIX

COMANDOS MAIS UTILIZADOS

A. Visualizar as primeiras linhas de um arquivo

```
head dmel-gene.fasta
```

B. Ver as 20 primeiras linhas do arquivo

```
head -20 dmel-gene.fasta
```

C. Visualizar as últimas linhas de um arquivo

```
tail dmel-gene.fasta
```

UNIX

COMANDOS MAIS UTILIZADOS

Visualização de arquivos (e informações)

- grep #busca de padroes

```
grep "padrao" input
```

```
grep ">" meta_coral.fasta
```

UNIX

COMANDOS MAIS UTILIZADOS

Visualização de arquivos (e informações)

- grep #busca de padroes

```
grep "padrao" input
```

```
grep "A" 454meta_coral.fasta
```

UNIX

COMANDOS MAIS UTILIZADOS

Visualização de arquivos (e informações)

- grep #busca de padroes

```
grep "padrao" input
```

```
grep "^A" meta_coral.fasta
```

UNIX

COMANDOS MAIS UTILIZADOS

Visualização de arquivos (e informações)

- grep #busca de padroes

```
grep "padrao" input
```

```
grep -c "^A" meta_coral.fasta
```

UNIX

grep - aplicações

Contar o número de sequências no arquivo

Número de sequências com o barcode "ACACGACGACT"

Número de sequências com o barcode "ACACGTAGTAT"

UNIX

grep - aplicações

Contar o número de sequências no arquivo

```
grep -c ">" ~/myseq/meta_coral.fasta
```

Número de sequências com o barcode "ACACGACGACT"

```
grep -c "ACACGACGACT" ~/myseq/meta_coral.fasta
```

Número de sequências com o barcode "ACACGTAGTAT"

```
grep -c "ACACGTAGTAT" ~/myseq/meta_coral.fasta
```

UNIX

grep - aplicações

Contar o número de sequências no arquivo

```
grep -c ">" ~/myseq/meta_coral.fasta
```

Número de sequências com o barcode "ACACGACGACT"

```
grep -c "^ACACGACGACT" ~/myseq/meta_coral.fasta
```

Número de sequências com o barcode "ACACGTAGTAT"

UNIX

grep - aplicações

Contar o número de sequências no arquivo

```
grep -c ">" ~/myseq/meta_coral.fasta
```

Número de sequências com o barcode "ACACGACGACT"

```
grep -c "^ACACGACGACT" ~/myseq/meta_coral.fasta
```

Número de sequências com o barcode "ACACGTAGTAT"

```
grep -c "^ACACGTAGTAT" ~/myseq/meta_coral.fasta
```

UNIX

COMANDOS MAIS UTILIZADOS

Redirecionando saída (e informações)

- >
- >>

```
grep '^>' input >output
```

```
grep '^>' meta_coral.fasta >tmp.fasta
```

UNIX

COMANDOS MAIS UTILIZADOS

Redirecionando saída (e informações)

- >
- >>

```
grep '^>' input >output
```

```
grep '^A' meta_coral.fasta >tmp.fasta
```

UNIX

COMANDOS MAIS UTILIZADOS

Redirecionando saída (e informações)

- >
- >>

```
grep '^>' input >output
```

```
grep '^>' meta_coral.fasta >>tmp.fasta
```

UNIX

EXEMPLOS

Trocar as letras maiúsculas por minúsculas

```
tr [A-Z] [a-z] <meta_coral.fasta >tmp.fasta
```

Fazer o complemento da sequência

UNIX

EXEMPLOS

Trocar as letras maiúsculas por minúsculas

```
tr [A-Z] [a-z] <meta_coral.fasta >tmp.fasta
```

Fazer o complemento da sequência

```
tr [ACGT] [tgca] <meta_coral.fasta >tmp.fasta
```

UNIX

COMANDOS MAIS UTILIZADOS

Edição de arquivos

- sed # stream editor

USO I:

```
sed s/texto_antigo/texto_novo/ <input>output
```

```
sed s/length/comprimento/ <meta_coral.fasta>tmp.fasta
```

UNIX

COMANDOS MAIS UTILIZADOS

Edição de arquivos

- sed # stream editor

USO I:

```
sed s/texto_antigo/texto_novo/flag <input >output
```

```
sed s/length/comprimento/ <meta_coral.fasta >tmp.fasta
```

FLAGS: "N" substitui até a enésima ocorrência
"g" substitui todas as ocorrências

UNIX

CONTINUAÇÃO

Edição de arquivos

- sed # stream editor

USO II:

sed y/caracteres/caracteres/ <input>output

```
sed y/ACGT/tgca/ <meta_coral.fasta>tmp.fasta
```

UNIX

COMANDOS MAIS UTILIZADOS

Edição de arquivos

- sed # stream editor

USO III:

sed G <input>output

```
sed G <meta_coral.fasta>tmp.fasta
```

UNIX

COMANDOS MAIS UTILIZADOS

Edição de arquivos

- sed # stream editor

USO IV:

sed = <input>output

```
sed = <meta_coral.fasta>tmp.fasta
```

UNIX

COMANDOS MAIS UTILIZADOS

Pipelines

- " | " # a saída de um comando é usada por outro

USO:

comando1 parametros | comando2 parametros

```
echo 5+6 | bc
```

```
grep ">" meta_coral.fasta | wc -l
```

```
tr [ACGT] [tgca] <meta_coral.fasta |  
sed s/\>/\>REVSEQ\_/ >tmp.fasta
```

UNIX

COMANDOS MAIS UTILIZADOS

```
tar -xvzf prova1.tar.gz
```

```
head prova1.doc
```

UNIX

COMANDOS MAIS UTILIZADOS

Permissões:

```
Darwin:Aluno Tatiana$ ls -la
total 5088
drwxr-xr-x  5 Tatiana  staff      170 22 Ago 14:02 .
drwxr-xr-x 49 Tatiana  staff    1666 22 Ago 14:02 ..
-rw-r--r--  3 Tatiana  staff  2593130 20 Ago 20:20 meta_coral.fasta
-----  1 Tatiana  staff       8 22 Ago 11:38 prova1.doc
-rw-r--r--@ 1 Tatiana  staff     152 22 Ago 12:18 prova1.tar.gz
```



UNIX

COMANDOS MAIS UTILIZADOS

Permissões:

```
Darwin:Aluno Tatiana$ ls -la
total 5088
drwxr-xr-x  5 Tatiana  staff      170 22 Ago 14:02 .
drwxr-xr-x 49 Tatiana  staff    1666 22 Ago 14:02 ..
-rw-r--r--  3 Tatiana  staff  2593130 20 Ago 20:20 meta_coral.fasta
-----  1 Tatiana  staff       8 22 Ago 11:38 prova1.doc
-rw-r--r--@ 1 Tatiana  staff     152 22 Ago 12:18 prova1.tar.gz
```



UNIX

COMANDOS MAIS UTILIZADOS

Permissões:

drwxr-xr-x

drwxr-xr-x

-rw-r--r--

-rw-r--r--



UNIX

COMANDOS MAIS UTILIZADOS

Permissões:

| u | g | o |
|------------|---|---|
| drwxr-xr-x | | |
| drwxr-xr-x | | |
| -rw-r--r-- | | |
| -rw-r--r-- | | |



UNIX

CONTINUAÇÃO

Mudando permissões

- chmod # change mode

PRIMEIRA FORMA (simbolica):

| OPERADOR | MEMÓRIA |
|----------|--------------------|
| + | Adiciona permissão |
| - | Remove permissão |
| = | Define permissão |

UNIX

CONTINUAÇÃO

Mudando permissões

- chmod # change mode

PRIMEIRA FORMA:

```
$ ls -l prova1.doc
```

UNIX

CONTINUAÇÃO

Mudando permissões

- chmod # change mode

PRIMEIRA FORMA:

```
$ ls -l prova1.doc
----- 1 Tatiana staff 8 22 Ago 2012 prova1.doc
```

UNIX

CONTINUAÇÃO

Mudando permissões

- chmod # change mode

PRIMEIRA FORMA:

```
$ ls -l prova1.doc
----- 1 Tatiana staff 8 22 Ago 2012 prova1.doc
$ chmod o+wr prova1.doc
$ ls -l prova1.doc
```

UNIX

CONTINUAÇÃO

Mudando permissões

- chmod # change mode

PRIMEIRA FORMA:

```
$ ls -l prova1.doc
----- 1 Tatiana staff 8 22 Ago 2012 prova1.doc
$ chmod o+wr prova1.doc
$ ls -l prova1.doc
-----rw- 1 Tatiana staff 8 22 Ago 2012 prova1.doc
```

UNIX

CONTINUAÇÃO

Mudando permissões

- chmod # change mode

PRIMEIRA FORMA:

```
$ chmod u+x prova1.doc  
$ ls -l prova1.doc
```

UNIX

CONTINUAÇÃO

Mudando permissões

- chmod # change mode

PRIMEIRA FORMA:

```
$ chmod u+x prova1.doc
$ ls -l prova1.doc
----x---rwx- 1 Tatiana staff 8 22 Ago 2012 prova1.doc
$ chmod g=rx prova1.doc
$ ls -l prova1.doc
---xr-xrw- 1 Tatiana staff 8 22 Ago 2012 prova1.doc
```

UNIX

CONTINUAÇÃO

Mudando permissões

- chmod # change mode

PRIMEIRA FORMA:

```
$ chmod u+rwx prova1.doc
$ ls -l prova1.doc
-rwxr-xrw-@ 1 Tatiana  staff  8 22 Ago  2012 prova1.doc
```

UNIX

CONTINUAÇÃO

Mudando permissões

- chmod # change mode

PRIMEIRA FORMA:

```
$ chmod u+rwx prova1.doc
```

UNIX

COMANDOS MAIS UTILIZADOS

Permissões:

“-rw-r--r--r” é equivalente a “644”

r = 4, w = 2, x = 1

| | user | group | other |
|---------|------|-------|-------|
| read | rw- | r-- | r-- |
| write | 4 r | 4 r | 4 r |
| execute | 2 w | 0 - | 0 - |
| | 0 - | 0 - | 0 - |
| soma | 6 | 4 | 4 |

UNIX

CONTINUAÇÃO

Mudando permissões

- chmod # change mode

SEGUNDA FORMA:

```
$ chmod 666 prova1.doc
$ ls -l prova1.doc
-rw-rw-rw-@ 1 Tatiana  staff  8 22 Ago  2012 prova1.doc
```

UNIX

CONTINUAÇÃO

Mudando permissões

- chmod # change mode

SEGUNDA FORMA:

```
$ chmod 666 prova1.doc
$ ls -l prova1.doc
-rw-rw-rw-@ 1 Tatiana  staff  8 22 Ago  2012 prova1.doc
```

UNIX

CONTINUAÇÃO

Mudando permissões

- chmod # change mode

SEGUNDA FORMA:

```
$ chmod 000 prova1.doc
$ ls -l prova1.doc
----- 1 Tatiana staff 8 22 Ago 2012 prova1.doc
```

UNIX

CONTINUAÇÃO

Super user do

- sudo

sudo command

```
$ sudo head prova1.doc  
Password:
```

Introdução à Programação de Computadores para Biologia

Algoritmos

ALGORITMOS

O que são?

"Uma série de instruções para a resolução de um problema"

ALGORITMOS

O que são?

"Uma série de instruções para a resolução de um problema"

- Exemplo I:
 - Problema: acordar de manhã.

ALGORITMOS

O que são?

"Uma série de instruções para a resolução de um problema"

- Exemplo I:
 - Problema: acordar pela manhã.
 - Solução: descrição dos passos necessários para essa árdua tarefa!

ALGORITMOS

EXEMPLO I: Algoritmo matinal

1. Acordar;
2. Se o despertador não tocou,
 - 2.1) voltar a dormir;
3. Se o despertador tocou,
 - 3.1) apertar o botão soneca;
4. Repetir o passo 3 por "n" vezes;
5. Levantar e caminhar até o banheiro;
6. Escovar os dentes;
7. Ligar a cafeteira;
8. Se não houver uma xícara limpa;
 - 8.1) ir ao passo 19;
9. Se houver uma xícara limpa,
 - 8.2) Colocar água na xícara;
10. Adicionar a água na cafeteira;
11. Repetir os passos 9 e 10 até que o nível na cafeteira seja = a "l"
12. Pegar o café ou capsula de café;
13. Adicionar o café a cafeteira;
14. Repetir o passo 13 por "m" vezes, se não é uma cafeteira de cápsula;
15. Colocar a xícara embaixo da saída do café;
16. Apertar o botão com o desenho da xícara;
17. Esperar 1 min até a saída completa do café;
18. Saborear o café;
19. Desligar a cafeteira;
21. Repetir a função 6;
22. Sair;

ALGORITMOS

O que são?

"Uma receita para a resolução de um problema"

- Exemplo II:
 - Problema: preparar um bolo de cenoura
 - Solução: descrição da receita

ALGORITMOS

EXEMPLO II: Bolo de cenoura

1. Separar os seguintes ingredientes:
 - 1.1) 3cenouras médias raladas;
 - 1.2) 4 ovos;
 - 1.3) 1/2 xícara (chá) de óleo;
 - 1.4) 2 xícaras (chá) de açúcar;
 - 1.5) 2 1/2 xícaras (chá) de farinha de trigo;
 - 1.6) 1 colher (sopa) de fermento em pó;
2. Ligar o forno para pré aquecimento a 180oC;
3. Bater no liquidificador as cenouras, os ovos e o óleo;
4. Transferir o resultado para uma vasilha;
5. Juntar aos poucos a farinha e o açúcar;
6. Misturar bem;
7. Misturar o fermento suavemente com uma colher;
8. Se temperatura do forno estiver a 180oC:
 - 8.1) Coloque o bolo no forno;
9. Senão, esperar até atingir 180oC e repetir o passo 8;
10. Assar o bolo até que, ao espetar com um palito, o palito fique limpo;
11. Tirar do forno;
12. O bolo está pronto.

ALGORITMOS: O QUE SÃO?

EXEMPLO II: Bolo de cenoura

- Objetos de consumo (entrada):
 - cenouras
 - ovos
 - farinha
 - óleo
 - açucar
 - fermento
- Objetos de apoio (atores, executores):
 - vasilha
 - liquidificador
 - xícara
 - colher de sopa
 - fogão
 - palito
 - cozinheiro(a)

ALGORITMOS: O QUE SÃO?

EXEMPLO II: Bolo de cenoura

- Objeto produzido (saída):
 - bolo
- Objeto que descreve o processo(receita):
 - Algoritmo

ALGORITMOS

O que os caracteriza?

1. É formado por um texto finito;
 - receita dada.
2. É composto por instruções elementares;
 - elementar depende do contexto.
3. É uma receita metódica, passo-a-passo;
 - passo inicial;
 - passo(s) intermediário(s);
 - passo final.
4. Ao executar:
 - partindo de dados válidos, deve sempre terminar;
 - partindo de dados inválidos, pode produzir lixo, ou mesmo não terminar.

ALGORITMOS

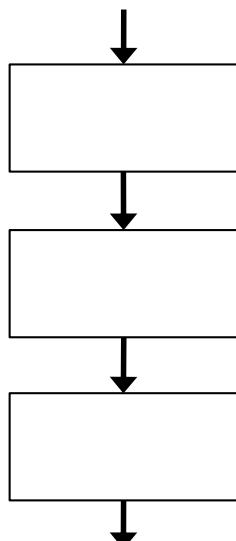
Representações

1. Linguagem escrita (português);

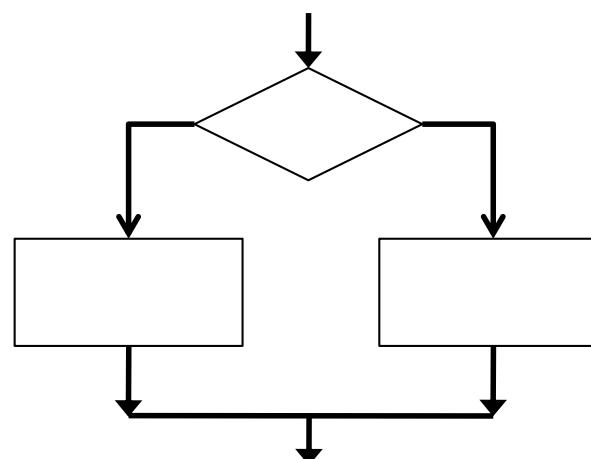
- exemplos I e II.

2. Fluxogramas;

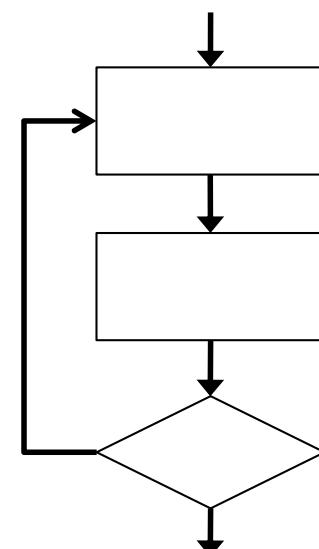
- representação gráfica.



Estrutura de controle
sequencial

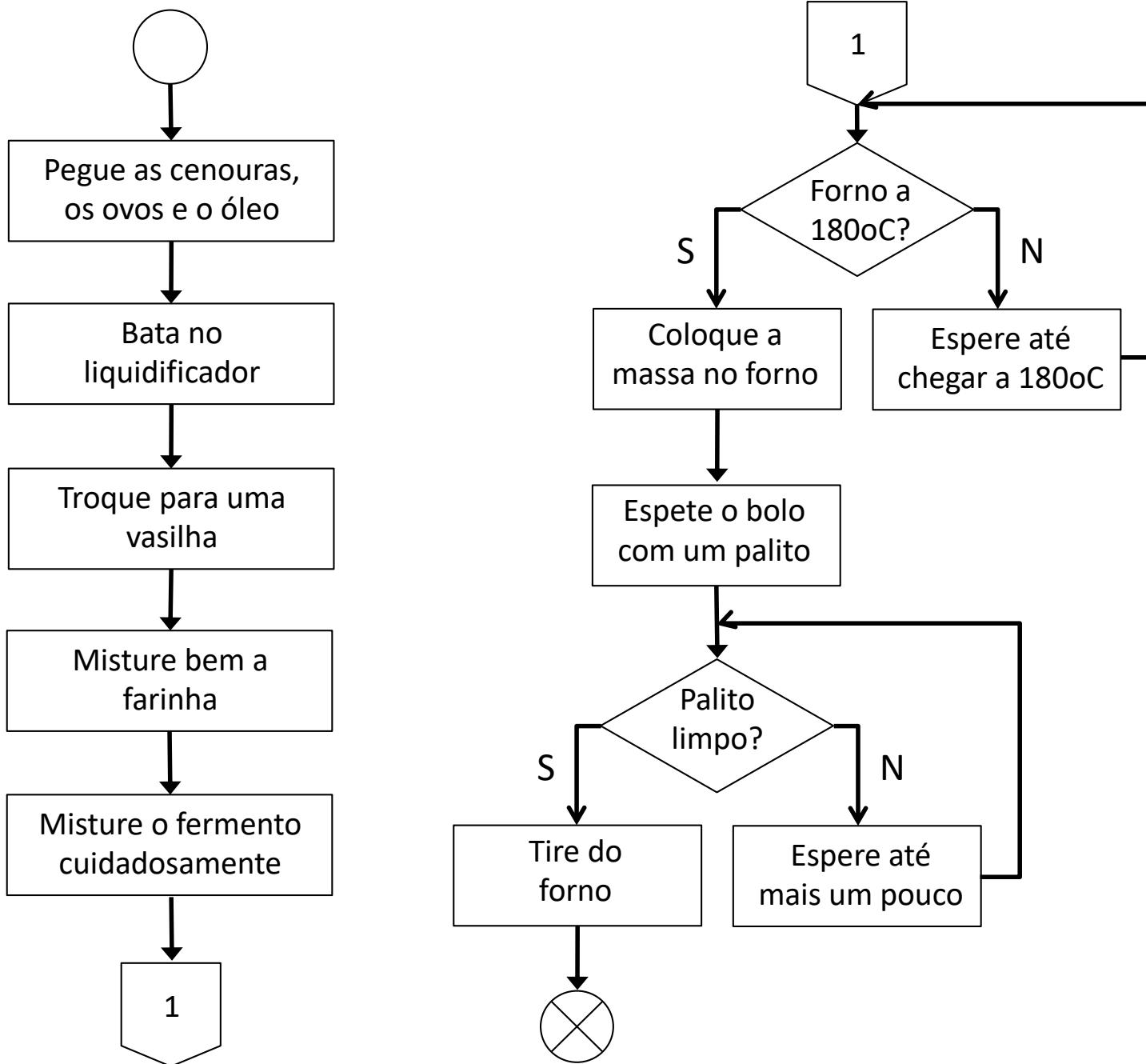


Estrutura de controle
condicional



Estrutura de controle
repetitiva

Representações: bolo de cenoura



ALGORITMOS

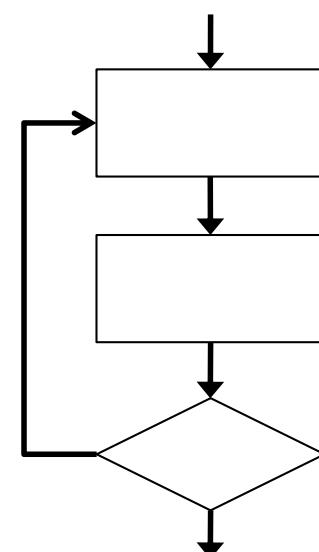
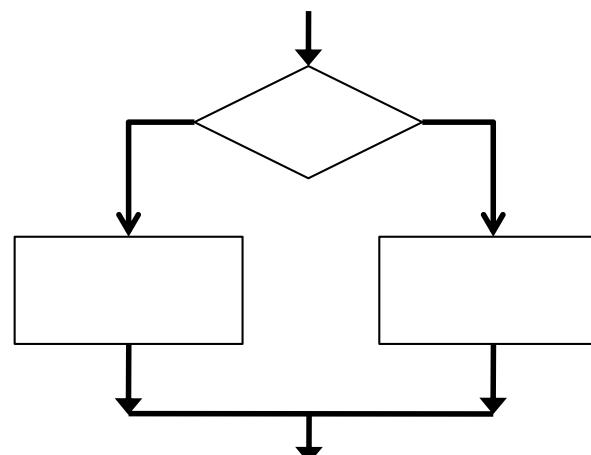
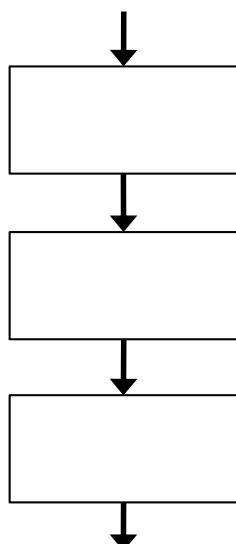
Representações

1. Linguagem escrita (português);

- exemplos I e II.

2. Fluxogramas;

- representação gráfica.



3. Linguagens de programação.

PERL

“Practical Extraction and Reporting Language”

PERL

“Practical Extraction and Reporting Language”

Larry Wall, "Pathologically Eclectic Rubbish Lister"

- Linguagem interpretada;
- muito utilizada em bioinformática;
- menos regras e múltiplas formas de resolver um problema.

PERL

“Practical Extraction and Reporting Language”

Larry Wall, "Pathologically Eclectic Rubbish Lister"

- Linguagem interpretada;
- muito utilizada em bioinformática;
- menos regras e múltiplas formas de resolver um problema.

“For programmers, laziness is a virtue.”

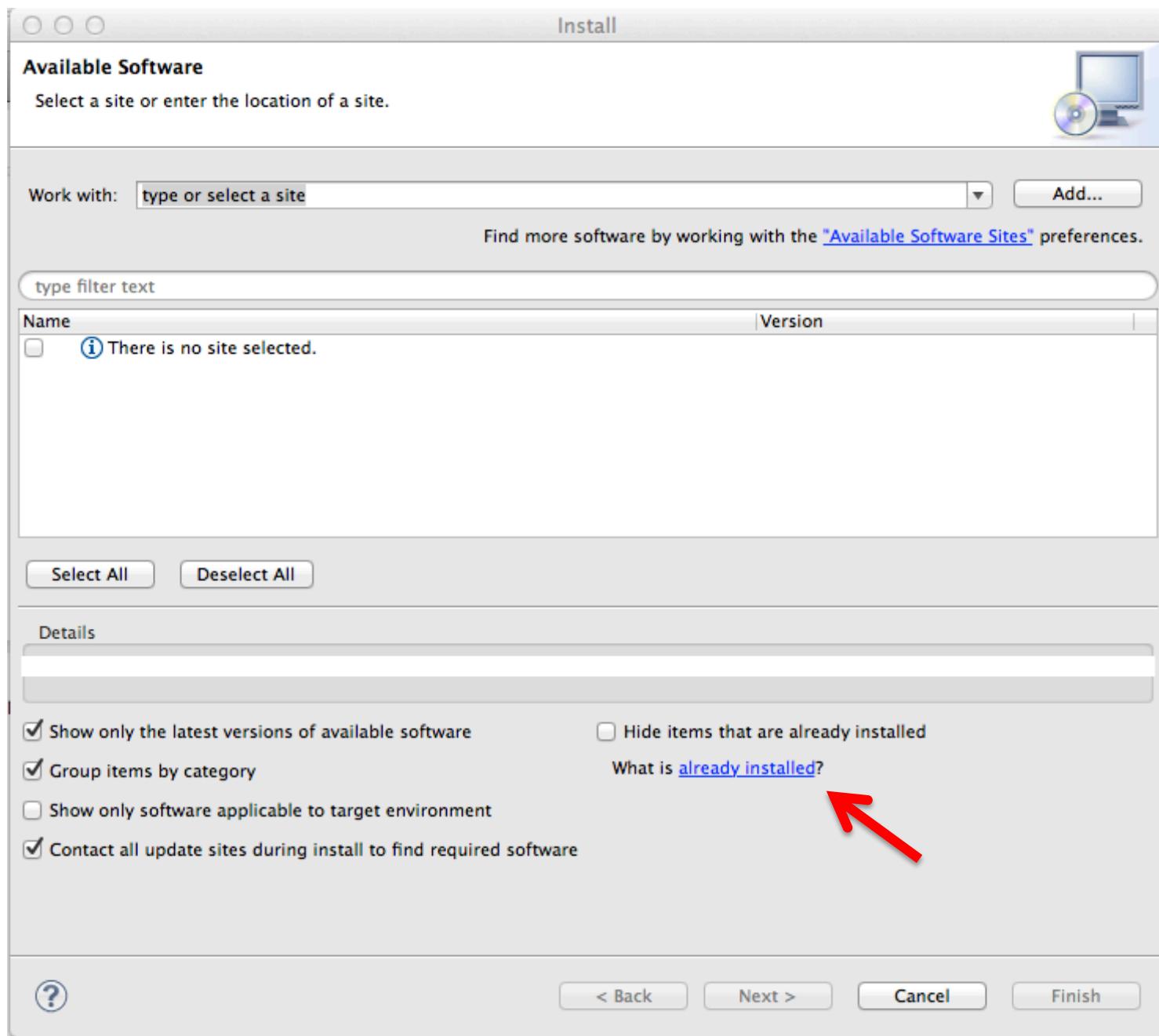


PERL

ECLIPSE E EPIC

- 1. Verificar a instalação do Eclipse;**
- 2. Se eclipse instalado, verificar a presença do EPIC**
 - Help > Install New Software...**
 - *What is already installed?***



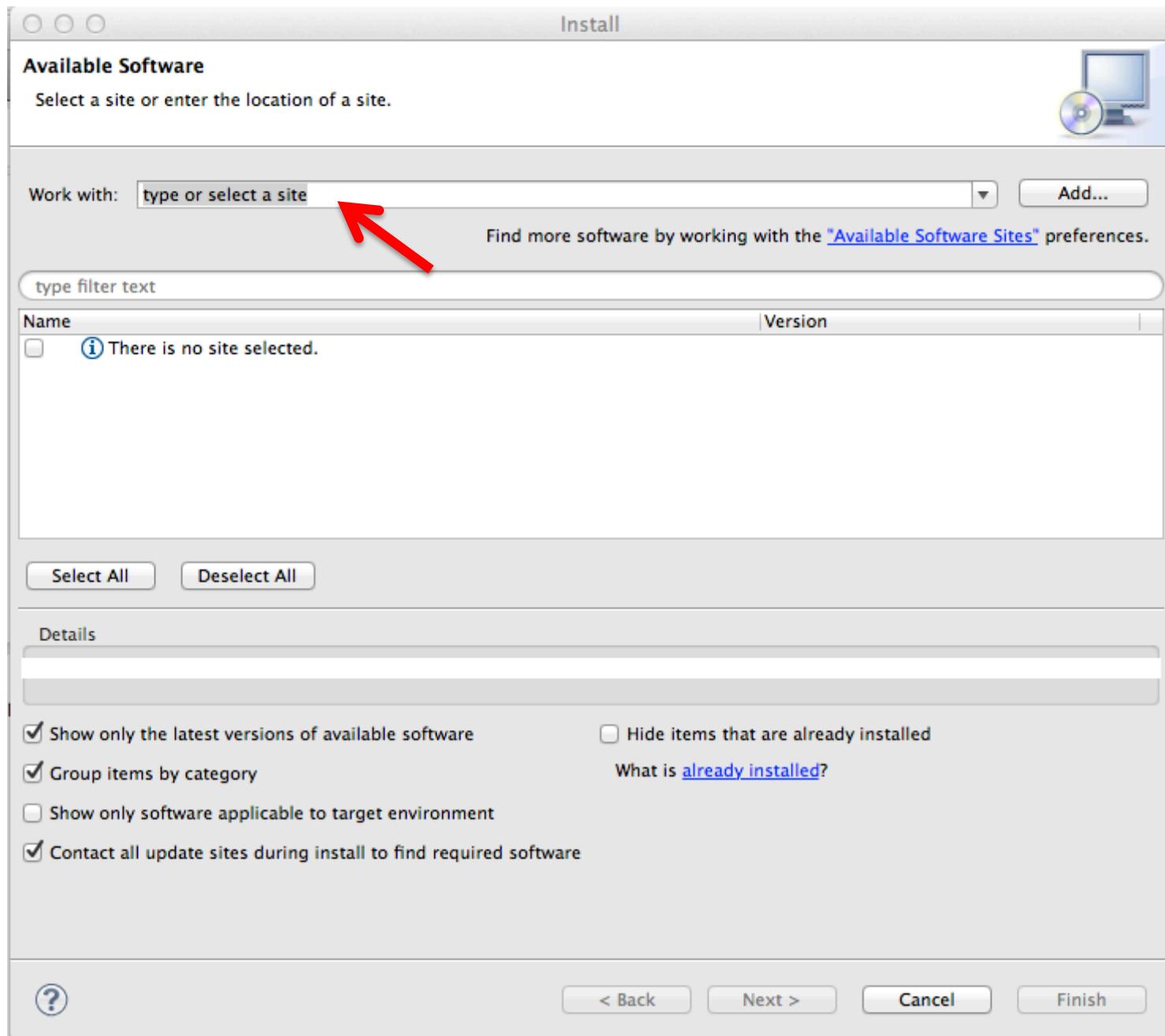


PERL

ECLIPSE E EPIC

1. Verificar a instalação do Eclipse;
2. Se eclipse instalado, verificar a presença do EPIC
 - Help > Install New Software...
 - *What is already installed?*
3. Se EPIC instalado, continuar a aula;
4. Se EPIC não instalado:
 - Help > Install New Software...
 - Adicionar o site <http://e-p-i-c.sf.net/updates/testing>





<http://e-p-i-c.sf.net/updates/testing>



PERL

ECLIPSE E EPIC

1. Verificar a instalação do Eclipse;
2. Se eclipse instalado, verificar a presença do EPIC
 - Help > Install New Software...
 - *What is already installed?*
3. Se EPIC instalado, continuar a aula;
4. Se EPIC não instalado:
 - Help > Install New Software...
 - Adicionar o site <http://e-p-i-c.sf.net/updates/testing>
 - Restart ECLIPSE



PERL

PRIMEIRO PROGRAMA

- 1. No Eclipse, *New > New Perl Project*;**
- 2. *New > New Perl File***
 - hello.pl

```
print "Hello world!\n";
```

-
- 3. Gravar arquivo**



PERL

PRIMEIRO PROGRAMA

4. No Terminal

```
p          e          r          l  
~/Documents/workspace/BI0514/hello.pl
```

5. No Eclipse

```
# Uhu! Meu primeiro script!  
print "Hello world!\n";
```

6. Rodar o script no Terminal.



PERL

PRIMEIRO PROGRAMA

4. No Terminal

```
perl  
~/Documents/workspace/BI0514/hello.pl
```

5. No Eclipse

```
# Uhu! Meu primeiro script!  
print "Hello world!\n";
```

6. Rodar o script no Terminal.



PERL

PRIMEIRO PROGRAMA

```
#! /usr/bin/perl  
  
# Uhu! Meu primeiro script!  
print "Hello world!\n";
```

```
chmod 777 hello.pl
```

```
ls -la
```

```
./hello.pl
```



ESTRUTURA DO CÓDIGO FONTE

```
#####
# PROGRAM: combine_tables.pl
# AUTHOR: Tatiana Torres
#
# LAST MODIFIED: 30.04.2010
#####
```

28.10.2009

Cabeçalho

```
#!/usr/bin/perl
```

Shebang

```
my ($E, $P, $M, $nota_final);
```

Declarações

```
($E, $P, $M) = @ARGV;
$nota_final = ((2*$E)+(3*$P)+(5*$M))/10;
if ($nota_final >= 5) {
    print "Aluno aprovado\n";
} else {
    print "Aluno reprovado\n";
}
```

Instruções

```
# conversao de nota para conceitos, apenas para a Pos-graduacao
```

Declarações

```
if ($nota_final < 5) {
    print "R, Reprovado, sem direito a credito\n";
} elsif ($nota_final <= 7.0) {
    print "C, Regular, com direito a credito\n";
} elsif ($nota_final <= 8.5) {
    print "B, Bom, com direito a credito\n";
} else {
    print "A, Excelente, com direito a credito\n";
}

exit;
```

Pontuação

