

# Introdução à Programação de Computadores para Biologia

## Estruturas de controle

Aula 06b

<https://tttorres.github.io/introprog2021/>

# ESTRUTURAS DE CONTROLE

## O que são?

Avaliação de um argumento:  
VERDADEIRO ou FALSO

- Exemplos:
  - O valor "100" é maior que o valor "30"?
  - A palavra "ATG" é diferente da palavra "ATT"?
  - A palavra "100" é maior que a palavra "30"?

# ESTRUTURAS DE CONTROLE

## O que é verdadeiro/falso em Perl?

Não verdadeiro:

- zero
- string vazia
- lista vazia
- undefined (undef)

# ESTRUTURAS DE CONTROLE

## O que é verdadeiro/falso em Perl?

1. No Geany, File > New File.
2. File > Save as...
3. Gravar arquivo como [controle.pl](#)
4. Copiar **exemplo02** da página da disciplina.

# ESTRUTURAS DE CONTROLE

## O que é verdadeiro/falso em Perl?

Script: [controle.pl](#)

Função *defined()*, exemplo01:

```
#!/usr/bin/perl
# script para testar a função defined

# criando variáveis

$scalar1 = 1;
$scalar2 = 0;
$scalar3 = "";
$scalar4;

$temp = defined($scalar1);
print "Resultado do defined = $temp";

exit;
```

# ESTRUTURAS DE CONTROLE

## O que é verdadeiro/falso em Perl?

Script: [controle.pl](#)

Função *undef()*, exemplo03:

```
$temp = defined($scalar1);  
print "Resultado do defined = $temp";  
  
# exemplo03  
# comandos testar a função undef  
  
$temp1 = defined($scalar1); ## TRUE  
print "Resultado do defined \($temp1\) = $temp1\n";  
  
$temp2 = undef($scalar1); ## FALSE  
print "Resultado do defined \($temp2\) = $temp2\n";  
  
exit;
```

# COMPARAÇÕES

## Operadores - Números e textos

1. Perl não tem os operadores:  $\neq$ ,  $\leq$ ,  $\geq$ ;
2. O sinal "=" é utilizado para atribuição;

```
#!/usr/bin/perl

# atribuicao

$scalar1 = 1;
$scalar2 = 0;

exit;
```

3. Operadores para números e texto são diferentes.

# COMPARAÇÕES

## Operadores - Números e textos

QUESTÃO	Números	"Strings"
\$a é igual a \$b?	\$a == \$b	\$a eq \$b
\$a é diferente de \$b?	\$a != \$b	\$a ne \$b
\$a é maior que \$b?	\$a > \$b	\$a gt \$b
\$a é maior ou igual a \$b?	\$a >= \$b	\$a ge \$b
\$a é menor que \$b?	\$a < \$b	\$a lt \$b
\$a é menor ou igual a \$b?	\$a <= \$b	\$a le \$b
\$a é diferente de \$b? Nesse caso, \$a é maior ou menor que \$b?	\$a <=> \$b	\$a cmp \$b



# ESTRUTURAS DE CONTROLE

## Operadores - Números e textos

Script: [controle.pl](#), exemplo04:

```
#!/usr/bin/perl

...

# testando operacoes matematicas
# comparacao 1
my $c = 5;
my $e = 5;
my $d = 20;

$temp = $c == $e;
print "A comparacao 1 eh $temp.\n";

exit;
```

# ESTRUTURAS DE CONTROLE

## Operadores - Números e textos

Script: [controle.pl](#), exemplo04:

```
#!/usr/bin/perl

...

# testando operacoes matematicas
# comparacao 1
my $c = 5;
my $e = 5;
my $d = 20;

$temp = $c == $e;
print "A comparacao 1 eh $temp.\n"; ## TRUE

exit;
```

# ESTRUTURAS DE CONTROLE

## Operadores - Números e textos

Script: [controle.pl](#), exemplo04:

```
#!/usr/bin/perl

...

# testando operacoes matematicas
# comparacao 2
my $c = 5;
my $e = 5;
my $d = 20;

$temp = $c == $d;
print "A comparacao 2 eh $temp.\n";

exit;
```

# ESTRUTURAS DE CONTROLE

## Operadores - Números e textos

Script: [controle.pl](#), exemplo04:

```
#!/usr/bin/perl

...

# testando operacoes matematicas
# comparacao 2
my $c = 5;
my $e = 5;
my $d = 20;

$temp = $c == $d;
print "A comparacao 2 eh $temp.\n"; ## FALSE

exit;
```

# ESTRUTURAS DE CONTROLE

## Operadores - Números e textos

Script: [controle.pl](#), exemplo04:

```
#!/usr/bin/perl

...

# testando operacoes matematicas
# comparacao 3
my $c = 5;
my $e = 5;
my $d = 20;

$temp = $c <= $d;
print "A comparacao 3 eh $temp.\n";

exit;
```

# ESTRUTURAS DE CONTROLE

## Operadores - Números e textos

Script: [controle.pl](#), exemplo04:

```
#!/usr/bin/perl

...

# testando operacoes matematicas
# comparacao 3
my $c = 5;
my $e = 5;
my $d = 20;

$temp = $c <= $d;
print "A comparacao 3 eh $temp.\n"; ## TRUE

exit;
```

# ESTRUTURAS DE CONTROLE

## Operadores - Números e textos

Script: [controle.pl](#), exemplo04:

```
#!/usr/bin/perl

...

# testando operacoes matematicas
# comparacao 4
my $c = 5;
my $e = 5;
my $d = 20;

$temp = $c <=> $d;
print "A comparacao 4 eh $temp.\n"; ## -1, 0, 1

exit;
```

# ESTRUTURAS DE CONTROLE

## Operadores - Números e textos

Script: [controle.pl](#), exemplo04:

```
#!/usr/bin/perl

...

# testando operacoes matematicas
# comparacao 5 - contexto de "string"
my $c = 5;
my $e = 5;
my $d = 20;

$temp = $c eq $d;
print "A comparacao 5 eh $temp.\n";

exit;
```



# ESTRUTURAS DE CONTROLE

## Operadores - Números e textos

Script: [controle.pl](#), exemplo04:

```
#!/usr/bin/perl

...

# testando operacoes matematicas
# comparacao 5 - contexto de "string"
my $c = 5;
my $e = 5;
my $d = 20;

$temp = $c eq $d;
print "A comparacao 5 eh $temp.\n"; ## FALSE

exit;
```

# ESTRUTURAS DE CONTROLE

## Operadores - Números e textos

Script: [controle.pl](#), exemplo04:

```
#!/usr/bin/perl

...

# testando operacoes matematicas
# comparacao 6 - contexto de "string"
my $c = 5;
my $e = 5;
my $d = 20;

$temp = $c gt $d;
print "A comparacao 6 eh $temp.\n";

exit;
```

# ESTRUTURAS DE CONTROLE

## Operadores - Números e textos

Script: [controle.pl](#), exemplo04:

```
#!/usr/bin/perl

...

# testando operacoes matematicas
# comparacao 6 - contexto de "string"
my $c = 5;
my $e = 5;
my $d = 20;

$temp = $c gt $d;
print "A comparacao 6 eh $temp.\n"; ## TRUE!!!

exit;
```

# ESTRUTURAS DE CONTROLE

## Operadores - Números e textos

Script: [controle.pl](#), exemplo04:

```
#!/usr/bin/perl

...

# testando operacoes matematicas
# comparacao 7 - contexto de "string"
my $c = 5;
my $e = 5;
my $d = 20;

$temp = $c cmp $d;
print "A comparacao 7 eh $temp.\n"; ## -1, 0, 1

exit;
```

# ESTRUTURAS DE CONTROLE

## Operadores - Números e textos

Script: [controle.pl](#), exemplo04:

```
#!/usr/bin/perl

...

# testando operacoes matematicas
# comparacao 8 - contexto de "string"
# ordem alfabética

$geneA = "FBgn2";
$geneB = "FBgn100";

$temp = $geneA cmp $geneB;
print "A comparacao 8 eh $temp.\n";

exit;
```

# ESTRUTURAS DE CONTROLE

## Operadores - Números e textos

Script: [controle.pl](#), exemplo04:

```
#!/usr/bin/perl

...

# testando operacoes matematicas
# comparacao 9 - contexto de "string"
# ordem alfabética

$geneA = "FBgn002";
$geneB = "FBgn100";

$temp = $geneA cmp $geneB;
print "A comparacao 9 eh $temp.\n";

exit;
```

# ESTRUTURAS DE CONTROLE

## Condicionais

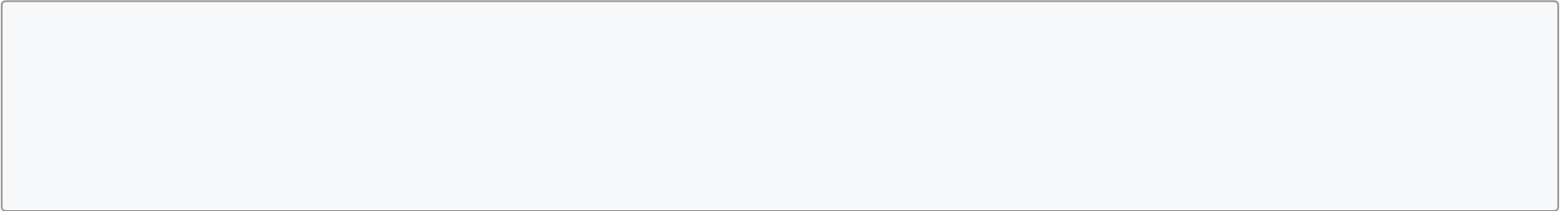
### Exemplo:

Determinar o tamanho de uma sequência somente SE a qualidade média dela for maior que 40.

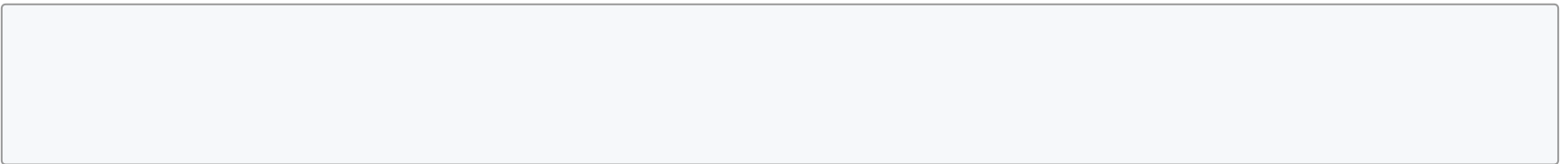
# ESTRUTURAS DE CONTROLE

## Condicionais

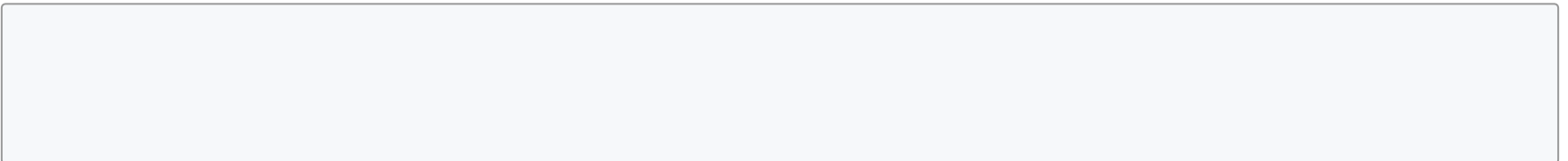
### 1. *if*

A large, empty rectangular box with a light blue background and a thin black border, intended for a code example of an if statement.

### 2. *boolean*

A large, empty rectangular box with a light blue background and a thin black border, intended for a code example of a boolean statement.

### 3. *else e elsif*

A large, empty rectangular box with a light blue background and a thin black border, intended for a code example of an else or elsif statement.



# ESTRUTURAS DE CONTROLE

## Condicionais

### 1. *if*

```
if (expressao) {  
    bloco de comandos  
}
```

### 2. *boolean*

### 3. *else e elsif*

# CONDICIONAIS

if, boolean, else

notas.pl

```
#!/usr/bin/perl

my ($E, $M, $H, $nota_final);

($E, $M, $H) = @ARGV;

$nota_final = ((2*$E)+(3*$M)+(5*$H))/10;

if ($nota_final >= 5) {
    print "Aluno aprovado\n";
} else {
    print "Aluno reprovado\n";
}

exit;
```

# CONDICIONAIS

## if, boolean, else

1. No Geany, File > New File.
2. File > Save as...
3. Gravar arquivo como [condicionais.pl](#)
4. Copiar **exemplo05** da página da disciplina e colar no script [condicionais.pl](#).

# CONDICIONAIS

if

**condicionais.pl**, exemplo 05

```
# exemplo05
#! /usr/bin/perl
# script para testar estruturas de controle

$seq = "ATGCGCGTAGCTCGTAGCTAGCTAGCTAGCT";
$avg_qual = 40;

if ($avg_qual > 30) {

    $seq_length = length($seq);

}

print "O tamanho da sequencia eh $seq_length.\n";

exit;
```

# CONDICIONAIS

if

## condicionais.pl, exemplo 05

```
# exemplo05
#! /usr/bin/perl
# script para testar estruturas de controle

$seq = "ATGCGCGTAGCTCGTAGCTAGCTAGCTAGCT";
$avg_qual = 40;

if ($avg_qual > 40) {

    $seq_length = length($seq);

}

print "O tamanho da sequencia eh $seq_length.\n";

exit;
```

# CONDICIONAIS

if

## condicionais.pl, exemplo 05

```
# exemplo05
#! /usr/bin/perl
# script para testar estruturas de controle

$seq = "ATGCGCGTAGCTCGTAGCTAGCTAGCTAGCT";
$avg_qual = 40;

if ($avg_qual >= 40) {

    $seq_length = length($seq);

}

print "O tamanho da sequencia eh $seq_length.\n";

exit;
```

# CONDICIONAIS

if

**condicionais.pl**, exemplo 06

```
# exemplo06
# comando die()

$nota_do_aluno = 4.5;
$nota_de_corte = 5.0;

if ($nota_do_aluno < $nota_de_corte) {
    die "Aluno reprovado\n";
}

exit;
```

# CONDICIONAIS

if

**condicionais.pl**, exemplo 06

```
# exemplo06
# comando die()

$nota_do_aluno = 4.5;
$nota_de_corte = 5.0;

if ($nota_do_aluno < $nota_de_corte) {
    die "Aluno reprovado\n";
}

print "Imprimindo qualquer coisa depois de die()";

exit;
```



# CONDICIONAIS

if

**condicionais.pl**, exemplo 06

```
# exemplo06
# comando die()

$nota_do_aluno = 5.5;
$nota_de_corte = 5.0;

if ($nota_do_aluno < $nota_de_corte) {
    die "Aluno reprovado\n";
}

print "Imprimindo qualquer coisa depois de die()";

exit;
```

# CONDICIONAIS

**if**

**condicionais.pl**, exemplo 07

```
# exemplo07  
# comando die()  
  
@array_vazio = ();  
  
$temp = pop(@array_vazio);  
  
if (! $temp) {  
    die "pop() nao deve ser usado em um array vazio\n";  
}  
  
exit;
```

# CONDICIONAIS

**if**

**Exemplo:**

Determinar se um mesmo elemento está em dois hashes diferentes e usar a informação desses hashes.

# CONDICIONAIS

if

## condicionais.pl, exemplo 08

```
# exemplo08
# comparando dois hashes

%coord_start = ("Gene1" => "100000",
                "Gene2" => "550000");
%coord_end   = ("Gene1" => "102000",
                "Gene2" => "556000");

$gene_alvo = "Gene1";

if ($coord_start{$gene_alvo}) {
    if ($coord_end{$gene_alvo}) {
        print "As coordenadas do $gene_alvo sao
              $coord_start{$gene_alvo}..$coord_end{$gene_alvo}. \n";
    }
}

exit;
```

# CONDICIONAIS

if

## condicionais.pl, exemplo 08

```
# exemplo08
# comparando dois hashes

%coord_start = ("Gene1" => "100000",
                "Gene2" => "550000");
%coord_end   = ("Gene2" => "556000");

$gene_alvo = "Gene1";

if ($coord_start{$gene_alvo}) {
    if ($coord_end{$gene_alvo}) {
        print "As coordenadas do $gene_alvo sao
               $coord_start{$gene_alvo}..$coord_end{$gene_alvo}. \n";
    }
}

exit;
```

# ESTRUTURAS DE CONTROLE

## Condicionais

### 1. *if*

```
if (expressao) {  
    bloco de comandos  
}
```

### 2. *boolean*

```
(expressao) && (expressao)  
(expressao) || (expressao)
```

### 3. *else e elsif*

# CONDICIONAIS

## Operadores booleanos

### condicionais.pl, exemplo 09

```
# exemplo09
# operadores booleanos &&, ||

%coord_start = ("Gene1" => "100000",
                "Gene2" => "550000");
%coord_end   = ("Gene1" => "102000",
                "Gene2" => "556000");

$gene_alvo = "Gene1";

if (($coord_start{$gene_alvo}) && ($coord_end{$gene_alvo})) {
    print "As coordenadas do $gene_alvo sao
    $coord_start{$gene_alvo}..$coord_end{$gene_alvo}. \n";
}

exit;
```

# CONDICIONAIS

## Operadores booleanos

### condicionais.pl, exemplo 09

```
# exemplo09
# operadores booleanos &&, ||

%coord_start = ("Gene1" => "100000",
                "Gene2" => "550000");
%coord_end   = ("Gene2" => "556000");

$gene_alvo = "Gene1";

if (($coord_start{$gene_alvo}) && ($coord_end{$gene_alvo})) {
    print "As coordenadas do $gene_alvo sao
    $coord_start{$gene_alvo}..$coord_end{$gene_alvo}. \n";
}

exit;
```



# CONDICIONAIS

## Operadores booleanos

### condicionais.pl, exemplo 09

```
# exemplo09
# operadores booleanos &&, ||

%coord_start = ("Gene1" => "100000",
                "Gene2" => "550000");
%coord_end   = ("Gene2" => "556000");

$gene_alvo = "Gene1";

if (($coord_start{$gene_alvo}) || ($coord_end{$gene_alvo})) {
    print "As coordenadas do $gene_alvo sao
    $coord_start{$gene_alvo}..$coord_end{$gene_alvo}. \n";
}

exit;
```

# ESTRUTURAS DE CONTROLE

## Condicionais

### 1. *if*

```
if (expressao) {  
    bloco de comandos  
}
```

### 2. *boolean*

```
(expressao) && (expressao)  
(expressao) || (expressao)
```

### 3. *else e elsif*

```
if (expressao) {  
    bloco de comandos  
} else {  
    bloco de comandos  
}
```

# CONDICIONAIS

## if, boolean, else

1. No Geany, File > New File.
2. File > Save as...
3. Gravar arquivo como [notas.pl](#)
4. Copiar **exemplo10** da página da disciplina e colar no script [notas.pl](#).

# CONDICIONAIS

## else e elsif

### notas.pl, exemplo 10

```
# exemplo 10
#! /usr/bin/perl
# if() e else()

$nota_do_aluno = 10.0;
$nota_de_corte = 5.0;

if ($nota_do_aluno < $nota_de_corte) {
    die "Aluno reprovado\n";
}

print "O script não morreu!\n";

exit;
```

# CONDICIONAIS

## else e elsif

### notas.pl, exemplo 10

```
# exemplo 10
#! /usr/bin/perl
# if() e else()

$nota_do_aluno = 4.0;
$nota_de_corte = 5.0;

if ($nota_do_aluno < $nota_de_corte) {
    die "Aluno reprovado\n";
}

print "O script não morreu!\n";

exit;
```

# CONDICIONAIS

## else e elsif

### notas.pl, exemplo 10

```
# exemplo 10
#! /usr/bin/perl
# if() e else()

$nota_do_aluno = 4.0;
$nota_de_corte = 5.0;

if ($nota_do_aluno < $nota_de_corte) {
    die "Aluno reprovado\n";
} # e se a nota for maior que 5?

print "O script não morreu!\n";

exit;
```

# CONDICIONAIS

## else e elsif

### notas.pl, exemplo 10

```
# exemplo 10
#! /usr/bin/perl
# if() e else()

$nota_do_aluno = 4.0;
$nota_de_corte = 5.0;

if ($nota_do_aluno < $nota_de_corte) {
    die "Aluno reprovado\n";
} # e se a nota for maior que 5?

if ($nota_do_aluno > $nota_de_corte) {
    print "Aluno aprovado\n";
}

exit;
```

# CONDICIONAIS

## else e elsif

### notas.pl, exemplo 10

```
# exemplo 10
#! /usr/bin/perl
# if() e else()

$nota_do_aluno = 6.0;
$nota_de_corte = 5.0;

if ($nota_do_aluno < $nota_de_corte) {
    die "Aluno reprovado\n";
} else {
    print "Aluno aprovado\n";
}

exit;
```



# CONDICIONAIS

## else e elsif

E se houver mais que duas alternativas?

**Exemplo:**

Conceitos na Pós-graduação

nota < 5.0, reprovado

nota <= 7.0, conceito C

nota <= 8.5, conceito B

nota > 8.5, conceito A

# CONDICIONAIS

## else e elsif

### notas.pl, exemplo 10

```
# exemplo 10
#! /usr/bin/perl
# if() e else()

$nota_do_aluno = 6.0;
$nota_de_corte = 5.0;

if ($nota_do_aluno < $nota_de_corte) {
    die "Aluno reprovado\n";
} else {
    if ($nota_do_aluno <= 7) {
        print "C, regular!\n";
    }
}

exit;
```

```
$nota_do_aluno = 6.0;
$nota_de_corte = 5.0;

print "\nExemplo 11:\n";

if ($nota_do_aluno < $nota_de_corte) {
    print "Aluno reprovado!\n";
} else {
    if ($nota_do_aluno <= 7) {
        print "C, regular!\n";
    } else {
        if ($nota_do_aluno <= 8.5) {
            print "B, bom!\n";
        } else {
            print "A, excelente!\n";
        }
    }
}

exit;
```

# else e elsif

## notas.pl, exemplo 11

```
$nota_do_aluno = 6.0;
$nota_de_corte = 5.0;

print "\nExemplo 10:\n";

if ($nota_do_aluno < $nota_de_corte) {
    print "Aluno reprovado!\n";
} elsif ($nota_do_aluno <= 7) {
    print "C, regular!\n";
} elsif ($nota_do_aluno <= 8.5) {
    print "B, bom!\n";
} else {
    print "A, excelente!\n";
}

exit;
```

# ESTRUTURAS DE CONTROLE

## Condicionais

### 1. *if*

```
if (expressao) {  
    bloco de comandos  
}
```

### 2. *boolean*

```
(expressao) && (expressao)  
(expressao) || (expressao)
```

### 3. *else e elsif*

```
if (expressao) {  
    bloco de comandos  
} else {  
    bloco de comandos  
}
```

