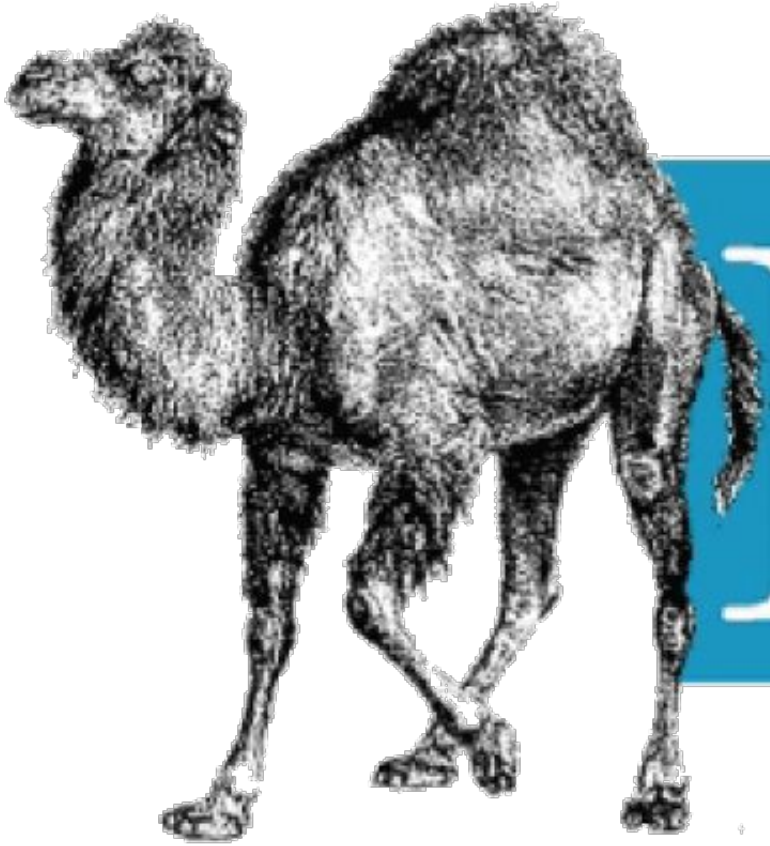


Tarefa 02

Perl

Felipe Oliveira
Renan Mendonça

Introdução

Perl

- Desenvolvido por Larry Wall em 1987
- Multiparadigma
- Multiplataforma
- Tipagem dinâmica
- Interpretada

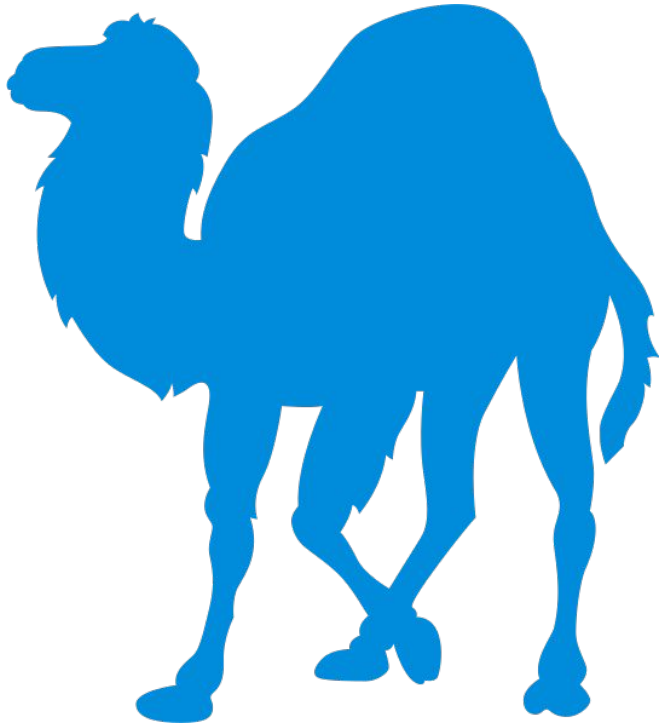
Origens e influências

- ‘sed’
- AWK
- C
- Shell Script

- Ruby on Rails

Usos

- Desenvolvimento de sites
- Computação gráfica
- Bioinformática
- Administração de sistemas
- Programação de redes
- Finança



Características

- Strings e expressões regulares
- Contexto de variáveis
  - Escalar e Lista
- Variável padrão $_
- Lento
- Maior writability
- Menor readability

Strings e expressões regulares

```
$str = "Paralisação na UERJ.";

if($str =~ /paralisação/i) {
    print "Tem paralisação.\n";
} else {
    print "Não tem paralisação.\n";
}
```

```
Tem paralisação.
```

```
$str = "O cachorro deitou no tapete.";
print "$str\n";

$str =~ s/cachorro/gato/;
print "$str\n";
```

```
O cachorro deitou no tapete.
O gato deitou no tapete.
```

```
$str = "O drone sobrevoou a UERJ.";
print "$str\n";

$str =~ tr/drone, UERJ/Drone, Uerj/;
print "$str\n";
```

```
O drone sobrevoou a UERJ.
O Drone sobrevoou a Uerj.
```

Contexto de variáveis

Escalar e Lista

```perl
@nomes = ('Rojas', 'Eustáquio');

@copia = @nomes;
$tamanho = @nomes;

print "Os nomes são: @copia\n";
print "Quantidade de nomes: $tamanho\n";
```

```
Os nomes são: Rojas Eustáquio
Quantidade de nomes: 2
```

```perl
my @linguagens = ('Pascal', 'Perl', 'Python');

if (@linguagens) {
    print "Existem itens nesta lista.\n";
} else {
    print "A lista está vazia.\n";
}
```

```
Existem itens nesta lista.
```

Variável padrão

$_

```perl
foreach('Larry ', 'Wall ') {
  print $_;
}
```

```
Larry Wall
```

```perl
@nomes = ('Huga ', 'Buga ');
for(@nomes) {
  print;
}
```

```
Huga Buga
```

```perl
my $arquivo = "employees.txt";

open my $fh, "<", $arquivo
  or die "Erro ao abrir: $_";

close $fh
  or die "Erro ao fechar: $_";
```

```
Erro ao abrir: at teste.pl line 2.
```

Avaliação comparativa

Concatenação de strings

```perl
# Entrada
my $str1 = <STDIN>;
my $str2 = <STDIN>;

# Remove as quebras de linha
chomp($str1, $str2);

# Concatena str2 à str1
$str1 = $str1 . $str2;

# Resultado
print $str1;
```

```c
#include <stdio.h>
int main() {
  char str1[100], str2[100], i, j;

  // Entrada
  scanf("%s", str1);
  scanf("%s", str2);

  // Calcula o tamanho da string str1
  for(i = 0; str1[i] != '\0'; ++i);

  // Concatena str2 à str1
  for(j = 0; str2[j] != '\0'; ++j, ++i) {
    str1[i] = str2[j];
  }

  // Adiciona o caracter nulo ao fim da string
  str1[i] = '\0';

  // Resultado
  printf("%s\n", str1);

  return 0;
}
```

Avaliação comparativa
Desempenho

Perl	C
● Interpretado ● Compila a fonte sempre que executado ● Procura módulos	● Compilado ● Fonte compilada apenas uma vez ● Mais eficiente