

Perl

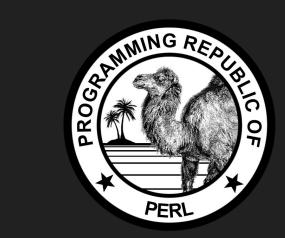
Trabalho 01 - Estruturas de Linguagem Luiz Henrique Andrade

Introdução

- Criada por Larry Wall em 1987
- Atualmente na versão 5 (5.24)
- Multiparadigma
 - Orientada a Objetos
 - Funcional
 - Imperativa
- "There's more than one way to do it"
- Multiplataforma
 - S.O. baseados em Unix já possuem interpretador Perl
- Tipagem Dinâmica

Influências

- C
- AWK
- SED
- Comandos Shell



```
Comparando com C
                                                           #!/usr/bin/perl
#include<stdio.h>
                                                           use strict;
int main()
                                                           use warnings;
  int n, first = 0, second = 1, next, c;
                                                           sub main {
  printf("Digite quantos numeros da sequência deseja ver\n");
  scanf("%d",&n);
                                                           print "\nDigite quantos numeros da sequência deseja ver\n";
                                                           my $n = \langle STDIN \rangle;
                                                           chomp $n;
  for ( c = 0 ; c < n ; c++ )
                                                           fibonacci($n);
                                                           exit 0;
     if ( c <= 1 )
        next = c;
                                                           sub fibonacci {
     else
                                                             $a = 0;
```

next = first + second;

first = second;

second = next;

printf("%d\n",next);

return 0;

b = 1;

my n = [0];

\$a = \$b; \$b = \$sum;

for (0...\$n-1){

printf "%d\n", \$a;
my \$sum = \$a + \$b;

```
#!/usr/bin/perl
                                   Comparando com Perl
use strict;
use warnings;
sub main {
print "\nDigite quantos numeros da sequência deseja ver\n";
my $n = \langle STDIN \rangle;
chomp $n;
fibonacci($n);
exit 0;
sub fibonacci {
 $a = 0;
 b = 1;
 my \ n = \ [0];
 for (0...$n-1){
   printf "%d\n", $a;
   my \$sum = \$a + \$b;
   a = b;
    $b = $sum;
```

```
use strict;
use warnings;
sub make fibonnaci {
    my ( $current, $next ) = (0, 1);
    return sub {
        my $fibonacci = $current;
        ( $current, $next ) = ( $next, $current + $next );
        return $fibonacci;
      };
    my $iterator = make fibonnaci();
    for (1 .. 10) {
      my $fibonacci = $iterator->();
```

print "\$fibonacci\n";