



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

```
#include<stdio.h>

int main()
{
    int n, first = 0, second = 1, next, c;
    printf("Digite quantos numeros da sequencia deseja ver\n");
    scanf("%d",&n);

    for ( c = 0 ; c < n ; c++ )
    {
        if ( c <= 1 )
            next = c;
        else
        {
            next = first + second;
            first = second;
            second = next;
        }
        printf("%d\n",next);
    }
    return 0;
}
```

```
#!/usr/bin/perl
use strict;
use warnings;

sub main {
    print "\nDigite quantos numeros da sequencia deseja ver\n";
    my $n = <STDIN>;
    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;
    }
}
```

Comparando com Perl

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

```
use strict;
```

```
use warnings;
```

```
sub main {
```

```
    print "\nDigite quantos numeros da sequência deseja ver\n";
```

```
    my $n = <STDIN>;
```

```
    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";
```

```
}
```