



Perl

# INTRODUÇÃO

- ▶ Criada por Larry Wall em 1987
- ▶ Multiplataforma
- ▶ Muita versatilidade
  - ▶ *"There's more than one way to do it"*
- ▶ Principal utilização na web
- ▶ Se destaca na manipulação de cadeias de caracteres

# ORIGENS E INFLUÊNCIAS


## Origens:

- ▶ C
- ▶ Snobol
- ▶ Lisp
- ▶ AWK
- ▶ SED

## Influências:

- ▶ Ruby on Rails

# CARACTERISTICAS

- ▶ Multiplataforma
  - ▶ Imperativa e Orientado a Objetos
  - ▶ Writeability (+)
  - ▶ Readability (-)
  - ▶ Lento
- 
- Several white lines of varying lengths and slopes are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

# CÓDIGO SIGNIFICATIVO

```
1  #!/usr/bin/perl
2
3  # checando qual o método usado para enviar os dados
4  if ($ENV{REQUEST_METHOD} eq 'POST')
5  { # se foi usado o método POST pegar os dados do STDIN
6    read(STDIN, $buffer, $ENV{CONTENT_LENGTH}); }
7  else
8  { # de outro modo pegar os dados da variável QUERY_STRING
9    $buffer = $ENV{QUERY_STRING}; }
10 # separando cada campo e seu valor
11 @pares = split(/&/, $buffer);
12 foreach $par (@pares)
13 { # jogando o nome do campo na variável campo e seu valor na variável valor
14   ($campo, $valor) = split("=", $par);
15   # convertendo + em espaço em branco
16   $valor =~ tr/+//g;
17   # convertendo os caracteres em hexadecimal para sua representação ASCII
18   $valor =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
19   # colocando no hash conteudo os valores dos campos, usando os nomes dos
20   # campos como chaves
21   $conteudo{$campo} = $valor;
22 }
23 # criando o pipe onde buscamos no arquivo cadastro o usuário e sua senha
24 open(BUSCA,"grep -x '$conteudo{usuario},$conteudo{senha}' cadastro |");
25 if (<BUSCA> ne "")
26 { # senha correta, enviando header para transmissão de arquivo binário
27   print "Content-Type: application/octet-stream\n\n";
28   # enviando o arquivo /usr/local/bin/transgif
29   open(ARQ,"/usr/local/bin/transgif");
30   while (<ARQ>) { print $_; } }
31 else
32 { # senha incorreta, enviando documento HTML com mensagem de erro
33   print "Content-type: text/html\n\n";
34   print "<h1>Usuário não cadastrado ou senha incorreta !\n" ]
35 close(BUSCA);
```

```
1  <form action=/cgi-bin/autenticacao.pl method=POST>
2
3      Nome: <input name=usuario>
4      Senha: <input type=password name=senha size=8>
5      <input type=submit>
6
7  </form>
```

# AVALIAÇÃO COMPARATIVA

```
1 #Imprimindo os elementos de um vetor em Pearl:
2 @list = (2, 20, 30, 40, 50);
3 foreach loop execution
4     foreach $a (@list){
5         print "value of a: $a\n";
6     }
```

```
1 //Imprimindo os elementos de um vetor em C:
2 struct reg {
3     int         conteudo;
4     struct reg *prox;
5 };
6 typedef struct reg celula;
7 void imprime (celula *le) {
8     if (le != NULL) {
9         printf ("value of a: %d\n", le->conteudo);
10        imprime (le->prox);
11    }
12 }
```