

Perl

{split e join}

15.1 JOIN

As duas operações que vamos aprender nesse capítulo são fundamentais para se trabalhar com arquivos no PERL e na Internet.

A primeira delas chama-se JOIN e serve para, tendo um array com vários elementos, montarmos uma variável em que estes elementos são "juntados" com um dado caractere entre eles (normalmente se usa dois-pontos). Assim, por exemplo:

```
@areg = ("aa","bb","cc");
$reg = join(":",@areg);
```

Em \$reg teremos **aa: bb: cc**

Se você rever a [aula 5](#) vai ver que um registro de arquivo em PERL é assim, por exemplo:

```
0001:José Silva:Rua C,36:01307-001:São Paulo:SP
```

onde temos os campos:Código do cliente, Nome, Endereço, CEP, Cidade e UF.

Você tem que se lembrar que isso não é SQL. Como dissemos naquela aula os campos em arquivos PERL não têm nome nem tipo. Terão de ser trabalhados pela sua posição na linha/registo do arquivo.

Repare que com o JOIN podemos, tendo um array com os valores de campos entrados numa página da Internet (vamos ver isso na prática no próximo capítulo), montar um registro de arquivo.

15.2 SPLIT

Já o operador SPLIT faz o contrário do JOIN. Ele pega um string com um separador se repetindo (dois-pontos no caso de registros) entre substrings e monta um array em que cada elemento é um destes substrings. Veja para entender:

```
$reg2 = "mn:op:qr";
@areg2 = split(/:/$reg2);
```

Teremos em @areg2 os elementos (mn,op,qr).

Na [aula 10](#) a gente viu como trabalhar com arrays. Você deve estar lembrado que podemos **captar** um elemento do array indexando assim:

```
$campotal = $areg2[1];
```

Na variável **campotal** teríamos o valor **op** (lembrando que a indexação começa com ZERO e que você tem que usar o símbolo \$ no nome do array).

É assim que poderemos ler uma linha/registo de um arquivo e montar os diversos campos, com os quais poderemos operar num programa.

Era mais ou menos assim que trabalhávamos na década de 60.Ô saudade!

É o que vamos ver mais em detalhes na próxima aula, a mais importante do curso!