05

O nosso primeiro problema

Transcrição

Já vimos um pequeno exemplo de regex e já sabemos que existem metacaracteres (*meta-char*) que possuem significados especiais, como o ponto (`.`) ou asterisco (`*`). No nosso primeiro exemplo para valor, vamos focar no CPF.

Uma tarefa muito comum no dia a dia do desenvolvedor é *parsear* um arquivo linha a linha, onde cada linha representa um dado ou registro. Há vários tipos de arquivos, e nesse curso vamos usar o exemplo de arquivo CSV, ou *Comma-separated Values*, por exemplo:

```
João Fulano,123.456.789-00,21 de Maio de 1993,(21) 3079-9987,Rua do Ouvidor,50,20040-030,Rio de Jane
Maria Fulana, 98765432100,11 de Abril de 1995,(11) 933339871,Rua Vergueiro,3185,04101-300,São Paulo
denise teste, 987.654.321.00,28 de Dezembro de 1991,(31)45562712,SCS Qd. 8 Bl. B-50,11,70333-900,Ric
```

Então, em cada linha temos vários valores separados pela vírgula, por exemplo:

```
João Fulano,123.456.789-00,21 de Maio de 1093,(21) 3079-9987,Rua Buarque de Macedo,67,22220-232,Rio
```

Encontrando números

Repare que o segundo valor é um CPF que precisamos extrair dessa linha. Você conhece um CPF e sabe o padrão de caracteres dele, só falta traduzir o seu conhecimento para a linguagem que a *regex engine* entende!

Um CPF são 9 números, separados em blocos de 3 números por um ponto, mais um hífen e mais dois números. Para representar um número, podemos utilizar uma *caracter class*, que é um símbolo especial para representar um conjunto de caracteres. No mundo de regex, um número é representado pelo `\d` .

O primeiro quantifier

Agora queremos que esse número apareça 3 vezes. Já vimos que o asterisco (`*`) significa 0, 1 ou mais vezes, ou seja, não atende. Queremos exatamente 3 números que podemos definir pela expressão `\d{3}` .

Dentro das chaves definimos a quantidade que o caractere deve estar presente. Com isso, já podemos encontrar 3 dígitos. Agora vem o "ponto" só que aprendemos que esse caractere possui um significado especial. No entanto queremos procurar o ponto literalmente e não qualquer caractere. Para deixar claro que o ponto deve ser ponto apenas, é preciso escapar o caractere com `\` . Assim temos:

```
\d{3}\.
```

Sabendo disso podemos definir o resto do CPF:

```
\d{3}\.\d{3}\.\d{3}\-\d{2}
```

Repare que o usamos um hífen seguido por apenas 2 dígitos.