

01

## Entendendo Classes de Caracteres

### Transcrição

No último capítulo demos uma introdução às regex e vimos nossa primeira expressão para encontrar um CPF em uma linha de texto. Já criamos uma expressão regular, bastante simples ainda encontrando o CPF:

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

Olhando novamente no nosso arquivo CSV, percebemos que há uma pequena variação da formatação do CPF:

987.654.321.00 e 11122233300 .

A nossa regex falha e não "enxerga" esses CPFs. Para fazer ela funcionar, devemos usar o poder das classes de caracteres. Na aula anterior já vimos uma classe predefinida, o `\d`. Essa classe na verdade é apenas um atalho pra `[0-9]`, ou seja qualquer dígito. Os colchetes definem uma classe de caracteres e nada impede que criemos nossa própria classe.

Repare que na décima segunda posição do CPF pode vir um ponto **OU** hífen, que podemos declarar com a classe: `[.-]`. Perceba também que não escapamos o ponto `(.)`, pois dentro dos colchetes o ponto já possui o seu valor literal. Importante também é que a ordem não importa, `[.-]` e `[-.]` são a mesma coisa.

### Caracteres opcionais

A quantidade de vezes que essa classe de caracteres deve aparecer é definida por um **quantifier**, por exemplo, como já vimos:

- `[.-]*` - ponto ou hífen zero, uma ou mais vezes.
- `[.-]{1}` - ponto ou hífen uma vez.

No nosso caso `[.-]` é opcional, pode ter ou não ter, mas uma vez apenas. Dentro da regex, isso é declarado através do *meta-char ponto de interrogação* (`?`). O ponto de interrogação (`?`), que significa *zero ou uma vez*, é mais um **quantifier**.

Assim podemos combinar a classe `[.-]` com o *quantifier* `?`:

- `[.-]?` - ponto ou hífen zero ou uma vez.

Vamos aplicar isso na nossa regex:

```
\d{3}[.-]?\d{3}[.-]?\d{3}[.-]?\d{2}
```

Agora sim, os CPFs serão encontrados com ou sem hífen/ponto.

Sabendo disso até podemos reescrever a classe `\d` (sim `\d` também é uma classe, já predefinida). Invés de usar `\d` podemos escrever:

```
[0123456789]
```

Ou mais curto:

[0-9]

Só cuidado, o hífen não possui tem um significado especial e define todos os números de 0 até 9.