

02

Validação customizada e mensagens em português

Já fomos capazes de aplicar algumas regras de validação em nosso formulário. Ao clicar em cadastrar sem preenchar nada ele mostra as mensagens de erro ao lado dos campos.

Mas e se eu desejo criar minha própria regra de validação? Por exemplo não desejo que nenhum produto possua a palavra "melhor", uma vez que a **melhor bola de futebol** poderia ser vendida e não ser de verdade a melhor de todas. Como dono da loja eu quero validar o campo nome com meu próprio validador.

Vamos então ao nosso controller, onde falamos que queremos validar o nome com:

```
$this->form_validation->set_rules("nome", "nome", "required");
```

Vamos agora pedir para ter tamanho mínimo 5:

```
$this->form_validation->set_rules("nome", "nome", "required|min_length[5]");
```

E também quero que ele não tenha a palavra melhor:

```
$this->form_validation->set_rules("nome", "nome", "required|min_length[5]|nao_tenha_a_p:
```

Como faço para criar o `nao_tenha_a_palavra_melhor`? Basta criar uma função em nosso controller:

```
public function nao_tenha_a_palavra_melhor() {  
}
```

Essa função recebe a string a ser validada:

```
public function nao_tenha_a_palavra_melhor($nome) {  
}
```

E verifico se ele encontra algo, qual a posição da palavra **melhor** lá dentro:

```
$posicao = strpos($nome, "melhor");
```

Caso a posição for diferente de `FALSE`, então quer dizer que não encontrou a palavra... retorna `TRUE`. Caso contrário deu erro:

```
if($posicao != FALSE) {  
    return TRUE;  
} else {  
    return FALSE;  
}
```

E tenho que adicionar a mensagem de erro no caso de algum problema de validação, com a categoria "nao_tenha_a_palavra_melhor" e a mensagem em si. Para escrever a mensagem posso substituir o nome do campo com %s :

```
public function nao_tenha_a_palavra_melhor($nome) {
    $posicao = strpos($nome, "melhor");
    if($posicao != FALSE) {
        return TRUE;
    } else {
        $this->form_validation->set_message("nao_tenha_a_palavra_melhor", "O campo '%s' não
        return FALSE;
    }
}
```

Salvamos e testamos com um produto como **melhor bola de futebol**, mas o teste passa, não dá erro! Se verificarmos, ainda não falamos pro code igniter que a `nao_tenha_a_palavra_melhor` não é igual a `required` e `min_length`. Esses dois últimos são padrões do Code Igniter enquanto o nosso é uma customização. Sendo assim, devemos dizer para ele invocar nossa função, invocar nosso callback na hora de validar:

```
$this->form_validation->set_rules("nome", "nome", "required|min_length[5]|callback_nao_
```

Agora sim testamos nossa aplicação e a mensagem customizada de erro aparece. O próximo passo é traduzir as outras mensagens para português: "The preco field is required" é feio para uma aplicação brasileira. Vamos entrar em um site que contem no github uma tradução de todas as mensagens de erro para o português e baixamos isso para nosso projeto:

<https://github.com/CIBr/CodeIgniter-Portuguese-BR> (<https://github.com/CIBr/CodeIgniter-Portuguese-BR>)

Clique em download as zip no projeto do Evaldo Junior, descompactamos o arquivo e dentro de `application/languages/` temos o diretório que devemos copiar, copie o diretório em si e cole em `application/languages` de sua aplicação (agora você tem tanto português quanto inglês).

No nosso arquivo de configuração `config.php` alteramos nossa `language` para `portuguese-br`, procure a configuração e altere para:

```
$config['language'] = 'portuguese-br';
```

Vimos como criar nosso próprio validador como um **callback**, como adicionar uma nova mensagem de erro e como configurar todas as mensagens do Code Igniter para o português.