05

Permit e relacionamentos

Capítulo 13 - *Permit* e relacionamentos

À medida em que a quantidade de produtos aumenta, começa a ficar difícil sua visualização e organização. Devemos arrumar um modo de agrupá-los em categorias. Como aqui estamos falando de vestimentas, podemos chamar cada categoria de "departamento". Com isso seremos capazes de fazer mudanças não apenas em um item, mas em vários ao mesmo tempo.

Scaffold

Até agora criamos o *controller*, o *model* e as páginas manualmente um por um. O *Rails* possui um comando que cria tudo de uma vez, o scaffold, que é a *estrutura* de cada departamento. Fazemos no console:

```
rails g scaffold Departamento nome
```

O Departamento será gerado com um nome característico, então passamos só essa variável. Não podemos esquecer de rodar as *migrations*:

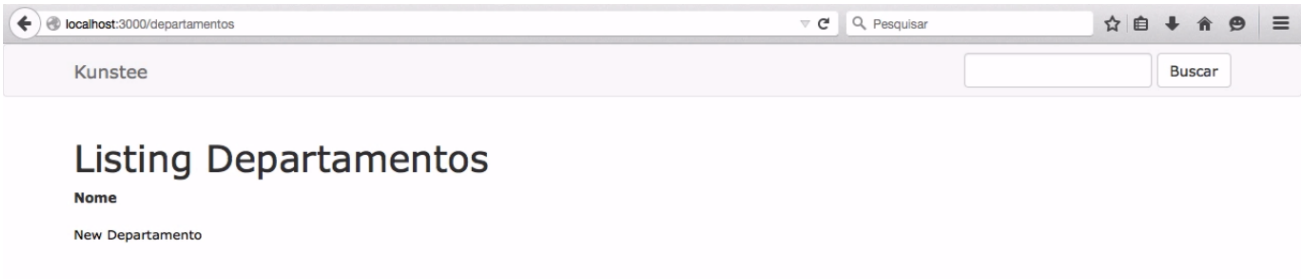
```
rake db:migrate
```

Assim é criado o arquivo dentro do diretório "/db/migrate":

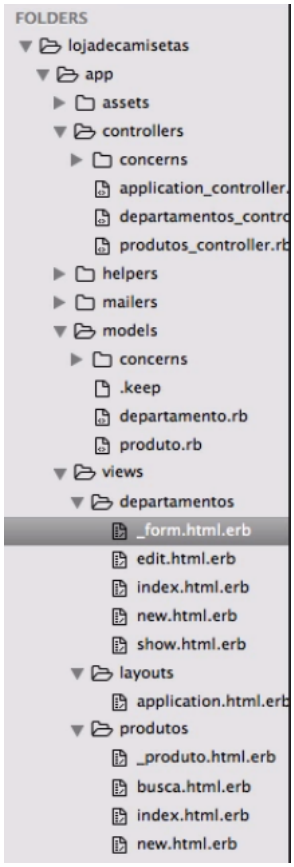
```
class CreateDepartamento < ActiveRecord::Migrtion
    def change
        create_table :departamentos do |t|
            t.string :nome

            t.timestamps null: false
        end
    end
end
```

Com isso foi criada a rota "localhost:3000/departamentos":



Nesta página já temos um link para criarmos um novo Departamento. Ele nos mostra um campo para darmos seu nome e um botão para efetivar a ação. Como podemos perceber, muita coisa foi criada automaticamente para nós: *models*, *routes*, *controllers* e páginas:



Muito disso não iremos usar ou é muito complexo e ainda precisamos implementar a parte gráfica usando o Bootstrap.

Relacionamento Produto-Departamento

Porém, o mais importante agora é efetivamente categorizar os produtos, ou seja, separá-los por departamento. Diremos que determinado produto pertence a determinado departamento e os relacionaremos no banco de dados. Para isto vamos ao arquivo "/db/schema". Lá encontraremos a nova tabela de departamentos e a de produtos:

```
create_table "departamentos", force: :cascade do |t|
  t.string   "nome"
  t.datetime "create_at", null: false
  t.datetime "updated_at", null: false
end

create_table "produtos", force: :cascade do |t|
  t.string   "nome"
  t.text     "descricao"
  t.integer  "quantidade"
  t.decimal  "preco"
  t.datetime "create_at", null: false
  t.datetime "update_at", null: false
end
```

O departamento deve ter uma *id* que fará essa relação com o produto, ou seja, na tabela de produtos, o departamento deverá aparecer referenciado. Vamos gerar uma nova migração. Existe uma convenção para gerar o SQL da tabela já configurado:

```
rails generate migration add_departamento_id_to_produto departamento_id:integer
```

Adicionamos (add) a coluna "departamento_id" na tabela do modelo "produto". Usamos o `integer` pois é um número que referenciará a chave do departamento. É criado um arquivo em "/db/migrate":

```
class AddDepartamentoIdProduto < ActiveRecord::Migration
    def change
        add_column :produtos, :departamento_id, :integer
    end
end
```

Rodando novamente a *migration*, é criada uma nova coluna no *schema*:

```
create_table "produtos", force: :cascade do |t|
  ...
  t.integer "departamento_id"
end
```

O produto pertence a um departamento, então fazemos em "/models/produto.rb"

```
class Produto < ActiveRecord::Base

    belongs_to :departamento

    validates ...
```

Quando invocamos esse método, estamos falando para o *Rails* a qual departamento um determinado produto pertence.

Antes de implementarmos a tela onde inseriremos o departamento do produto, façamos o mesmo no console primeiramente instanciando um produto (o primeiro da lista, no caso):

```
irb(main):001:0> big = Produto.first
... departamento_id: nil>
```

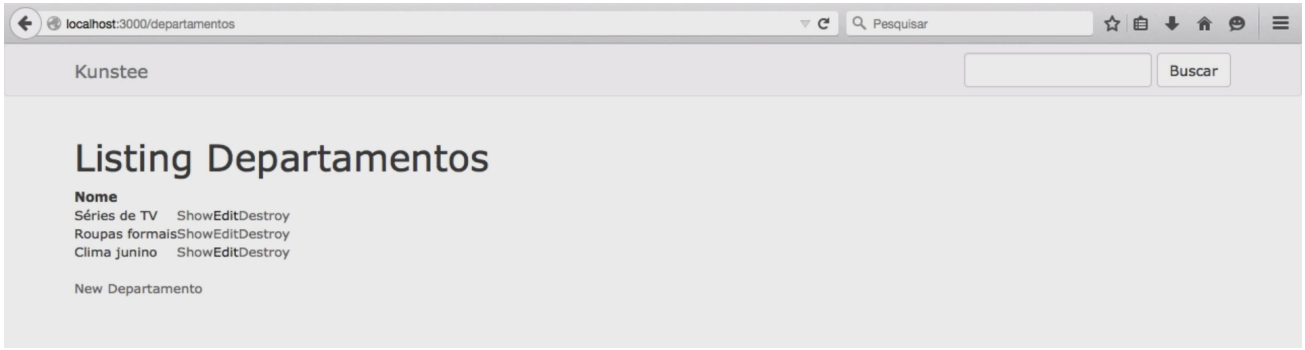
Perceba que o retorno já nos mostra a qual departamento o produto pertence, a nenhum até agora. Vamos inseri-lo no primeiro e único já criado, o "Séries de TV",

```
irb(main):002:0> big.departamento = Departamento.first
... #<Departamento id: 1, nome: "Séries de TV", ...
```

e salvá-lo:

```
irb(main):003:0> big.save
   (0.5ms)    begin transaction
    ... [["departamento_id", 1] ...
   (1.7ms)    commit transaction
=> true
```

O que queremos agora é criar um produto e já escolher seu departamento através da página de criação. Isso será feito por meio de um menu onde se escolhe a opção. Adicionamos algumas a mais:



No arquivo "/produtos/new.html.erb" adicionamos mais um campo usando o formulário do *Rails* (`f.label`):

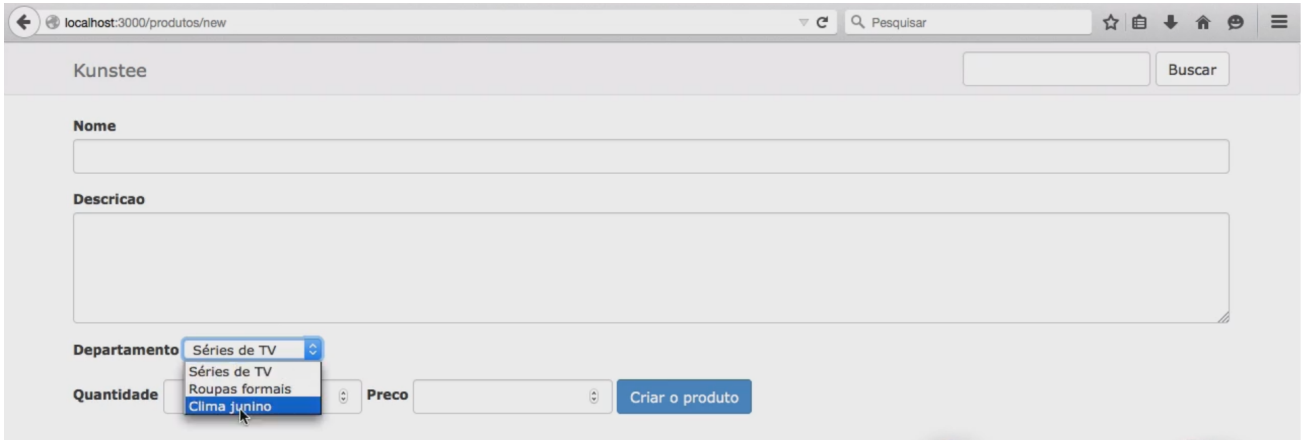
```
<div class="form-group">
    <%= f.label :departamento %>
    <%= f.collection_select :departamento_id, @departamentos, :id, :nome %>

</div>
```

Passamos qual o campo (do tipo `select`) do objeto que queremos popular (`departamento_id`) e precisamos buscar os departamentos (`@departamentos`). Também precisamos dizer qual campo de cada elemento do departamento que queremos invocar (`:id`) e qual nome queremos que apareça no `select` (`:nome`). E no *controller* fazemos:

```
def new
    @produto = Produto.new
    @departamentos = Departamento.all
end
```

Agora temos, na página de criação de produto, a opção de selecionar o departamento:

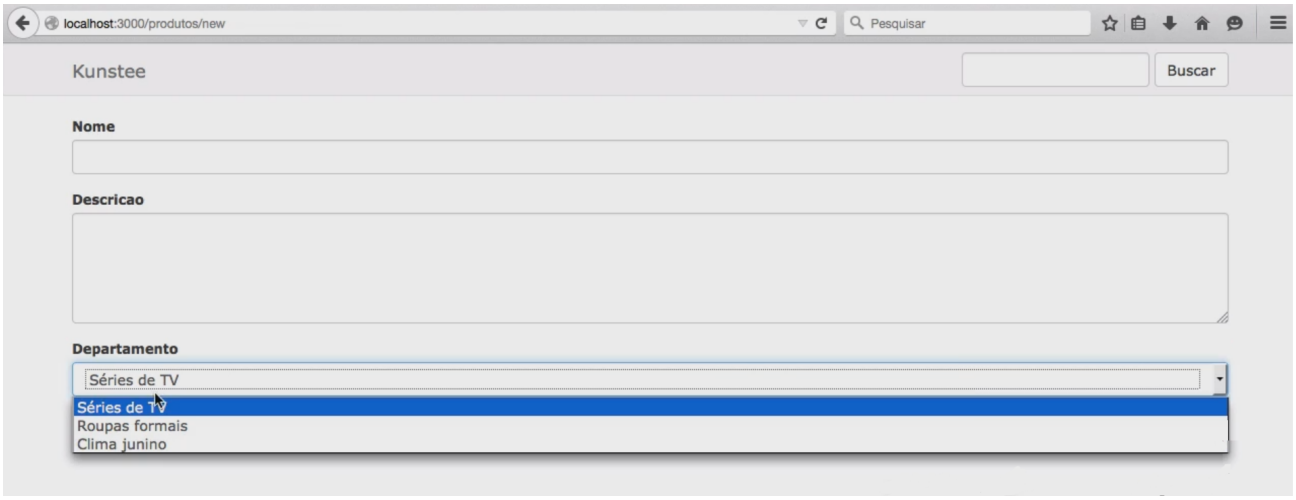


Bootstrap no departamento

Agora nos resta estilizar esse campo usando o Bootstrap (seção "Selects"):

```
<div class="form-group">
    <%= f.label :departamento %>
    <%= f.collection_select :departamento_id, @departamentos, :id, :nome, {}, class: "form-conti
</div>
```

Precisamos usar o *hash* `{}` para conseguirmos customizar o `select`. No final teremos essa cara para a seleção de departamentos:



Autorização do departamento_id

O próximo passo passo é fazer com que o servidor grave os dados. Do jeito que está o servidor irá gravar, mas mostrará um erro no console por falta de autorização:

```
... Unpermitted parameter: departamento_id
```

Precisamos colocar o parâmetro `departamento_id` na *whitelist* do produto:

```
def create
    valores = params.require(:produto).permit ..., :departamento_id
    ...
end
```

Agora temos permissão para salvar o departamento de cada produto. Podemos adicionar novos produtos e consultar seus departamentos via console para termos certeza que tudo está correndo bem.