

**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
RIO GRANDE DO NORTE
Campus Natal - Central

Enfeitando a Listagem

Prof. Fellipe Aleixo (fellipe.aleixo@ifrn.edu.br)

Iteração A2:

MELHORANDO A LISTAGEM

Atenção ao Feedback do Cliente

- O cliente fez alguns comentários:
 - “a listagem dos produtos está feia!”
 - “você pode melhora-la um pouco!”

Antes de Por a Mão na Massa

- Importante: os desenvolvedores envolvidos poderiam trabalhar com o mesmo conjunto de dados
- Rails oferece a possibilidade de importação de um conjunto inicial de dados (de teste) – “semente”
 - Alterar o arquivo “**seed.rb**” no diretório do banco

Populando o Banco de Dados

- Para popular a tabela é utilizado o método **create!()** da classe (de modelo) Produto

```
Download rails40/depot_a/db/seeds.rb
```

```
Product.delete_all
```

```
# . . .
```

```
Product.create!(title: 'Programming Ruby 1.9 & 2.0',
```

```
  description:
```

```
    %{\<p>
```

```
      Ruby is the fastest growing and most exciting dynamic language  
      out there. If you need to get working programs delivered fast,  
      you should add Ruby to your toolbox.
```

```
    </p>},
```

```
  image_url: 'ruby.jpg',
```

```
  price: 49.95)
```

```
# . . .
```

Populando o Banco de Dados

- O referido código utiliza `%{...}` como forma alternativa de representar constantes string
- O método **`create!()`** lança uma exceção caso a inserção não puder ser realizada
- Para popular a tabela é utilizado o comando:

```
depot> rake db:seed
```

Utilizando Imagens dos Produtos

- Para a utilização de imagens para os produtos as mesmas devem ser colocadas no diretório **app/assets/images**

Aspectos a Serem Melhorados

- Os melhoramentos da visualização podem ser divididos em duas tarefas:
 - Ajustar as definições de estilo (CSS)
 - Conectando tais definições à elementos da página
- Foi gerado o arquivo **products.css.scss** (vazio) no diretório **app/assets/stylesheets**

Download rails40/depot_a/app/assets/stylesheets/products.css.scss

```
// Place all the styles related to the Products controller here.  
// They will automatically be included in application.css.  
// You can use Sass (SCSS) here: http://sass-lang.com/
```

```
➤ .products {  
➤   table {  
➤     border-collapse: collapse;  
➤   }  
➤  
➤   table tr td {  
➤     padding: 5px;  
➤     vertical-align: top;  
➤   }  
➤  
➤   .list_image {  
➤     width: 60px;  
➤     height: 70px;  
➤   }  
➤  
➤   .list_description {  
➤     width: 60%;  
➤  
➤     dl {  
➤       margin: 0;  
➤     }  
➤   }  
➤ }
```

```
➤ dl {
➤   margin: 0;
➤ }
➤
➤ dt {
➤   color:      #244;
➤   font-weight: bold;
➤   font-size:  larger;
➤ }
➤
➤ dd {
➤   margin: 0;
➤ }
➤ }
➤
➤ .list_actions {
➤   font-size:  x-small;
➤   text-align: right;
➤   padding-left: 1em;
➤ }
```

```
➤ .list_line_even {
➤   background: #e0f8f8;
➤ }
➤
➤ .list_line_odd {
➤   background: #f8b0f8;
➤ }
➤ }
```

Aplicando as Regras de Estilo

- Rails mantém um arquivo separado para ser utilizado para definir um layout padrão para a aplicação
 - Arquivo: **application.html.erb**
 - Diretório: **layout**

Aplicando as Regras de Estilo

Download rails40/depot_a/app/views/layouts/application.html.erb

```
<!DOCTYPE html>
<html>
<head>
  <title>Depot</title>
  <%= stylesheet_link_tag "application", media: "all",
    "data-turbolinks-track" => true %>
  <%= javascript_include_tag "application", "data-turbolinks-track" => true %>
  <%= csrf_meta_tags %>
</head>
> <body class='<%= controller.controller_name %>'>
  <%= yield %>
</body>
</html>
```

```
<h1>Listing products</h1>
```

```
<table>
```

```
<% @products.each do |product| %>
```

```
  <tr class="<%= cycle('list_line_odd', 'list_line_even') %>">
```

```
    <td>
```

```
      <%= image_tag(product.image_url, class: 'list_image') %>
```

```
    </td>
```

```
    <td class="list_description">
```

```
      <dl>
```

```
        <dt><%= product.title %></dt>
```

```
        <dd><%= truncate(strip_tags(product.description), length: 80) %></dd>
```

```
      </dl>
```

```
    </td>
```

```
    <td class="list_actions">
```

```
      <%= link_to 'Show', product %><br/>
```

```
      <%= link_to 'Edit', edit_product_path(product) %><br/>
```

```
      <%= link_to 'Destroy', product, method: :delete,  
                data: { confirm: 'Are you sure?' } %>
```

```
    </td>
```

```
  </tr>
```

```
<% end %>
```

```
</table>
```

```
<br />
```




```
<%= link_to 'New product', new_product_path %>
```

Recursos Rails Utilizados

- `cycle()`
 - Permite alternar entre valores predefinidos
- `truncate()`
 - Utilizado para truncar um texto (reduzindo-o)
- `strip_tags()`
 - Remover as *tags* HTML do texto
- `link_to`
 - Definição de um link HTML

Resultado

Listing products

	CoffeeScript CoffeeScript is JavaScript done right. It provides all of JavaScript...	Show Edit Destroy
	Programming Ruby 1.9 & 2.0 Ruby is the fastest growing and most exciting dynamic language ...	Show Edit Destroy
	Rails Test Prescriptions Rails Test Prescriptions is a comprehensive guide to testing ...	Show Edit Destroy

[New product](#)