



BCC

Engenharia de Software

2018.2

Professor Rodrigo Andrade



O que é Rails?

- Framework de desenvolvimento Web
- Escrito em Ruby
- Visa facilitar o desenvolvimento de apps Web
- Segue o padrão MVC



http



Rack (Servidor de aplicação)



Rails

Três módulos principais

ActiveRecord

ActiveController

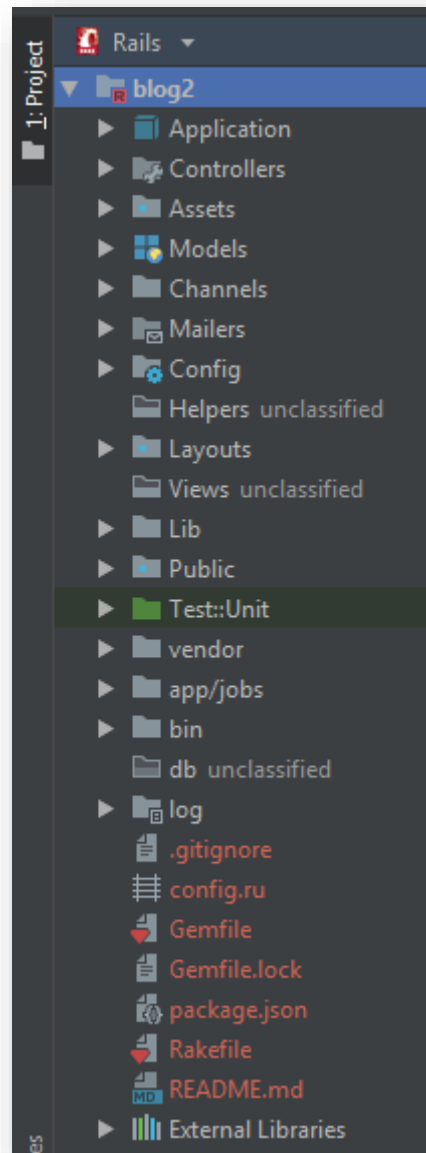
ActiveView

Estrutura geral de projetos Rails

Gemfile	lista de gemas Ruby (bibliotecas) que essa app usa (Capítulo 3)
Rakefile	comandos para automatizar manutenção e implantação (Capítulo 12)
app	sua aplicação
app/models	código do modelo
app/controllers	código dos controladores
app/views	modelos de visões
app/views/layouts	modelos de página usadas por todas as visões da app (vide text)
app/helpers	métodos de ajuda para modelos de visões
app/assets	itens estáticos (JavaScript, imagens, folhas de estilo)
config	informações básicas de configuração
config/environments	configurações para execução em desenvolvimento vs. produção
config/database.yml	configurações de bancos de dados (desenvolvimento vs. produção)

config/routes.rb	mapeamento de URIs para ações dos controladores
db	arquivos descrevendo o esquema dos bancos de dados
db/development.sqlite3	Armazenamento de dados para o banco de desenvolvimento SQLite
db/test.sqlite3	Banco de dados usado para executar os testes
db/migrate/	Migrações (descrições de mudanças no esquema do banco e dados)
doc	documentação gerada
lib	código adicional da app compartilhada entre M, V e C
log	arquivos de log
public	páginas de erro servidas pelo servidor Web
script	ferramentas de desenvolvimento, não parte da app
tmp	dados temporários mantidos em tempo de execução

Criando novo projeto



Rodando o servidor



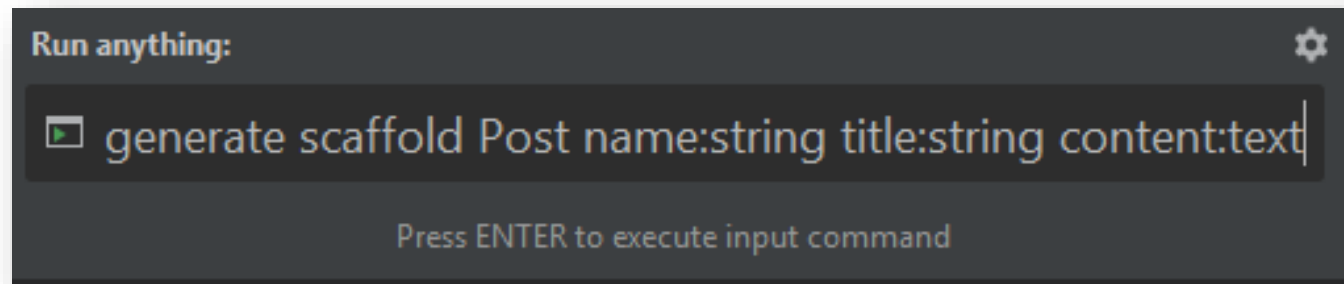
Yay! You're on Rails!



Rails version: 5.1.6

Ruby version: 2.3.3 (i386-mingw32)

Usando Scaffold



```
rails generate scafoold Post name:string title:string content:text
```

Rodando o servidor novamente

ActiveRecord::PendingMigrationError

Migrations are pending. To resolve this issue, run: `bin/rails db:migrate RAILS_ENV=development`


Extracted source (around line #576):

```
574     # Raises <tt>ActiveRecord::PendingMigrationError</tt> error if any migrations are pending.
575     def check_pending!(connection = Base.connection)
576       raise ActiveRecord::PendingMigrationError if ActiveRecord::Migrator.needs_migration?(connection)
577     end
578
579     def load_schema_if_pending!
```

Database migration

```
20181009135631_create_posts.rb x
1  class CreatePosts < ActiveRecord::Migration[5.1]
2    def change
3      create_table :posts do |t|
4        t.string :name
5        t.string :title
6        t.text :content
7
8        t.timestamps
9      end
10   end
11 end
```

Run anything:

```
 rake db:migrate
```

Rodando o servidor novamente

Sprockets::Rails::Helper::AssetNotFound in Posts#index

Showing *C:/ufrpe/es/workspace-ruby/blog2/app/views/layouts/application.html.erb* where line #7 raised:

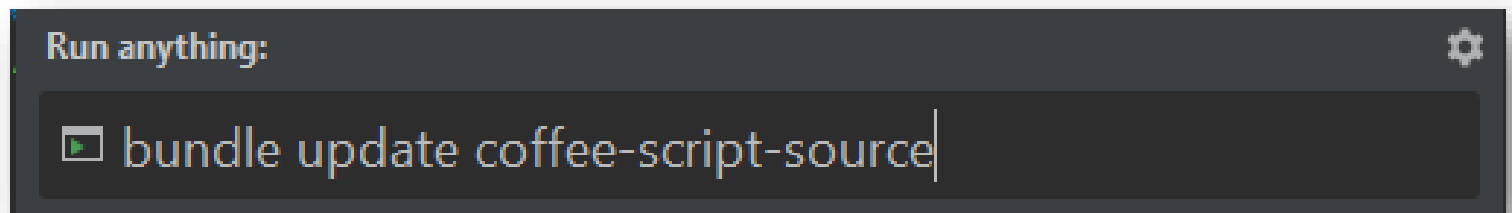
The asset "default.css" is not present in the asset pipeline.

Extracted source (around line #7):

```
5     <%= csrf_meta_tags %>
6
7     <%= stylesheet_link_tag 'default', media: 'all', 'data-turbolinks-track': 'reload' %>
8     <%= javascript_include_tag 'default', 'data-turbolinks-track': 'reload' %>
9 </head>
10
```

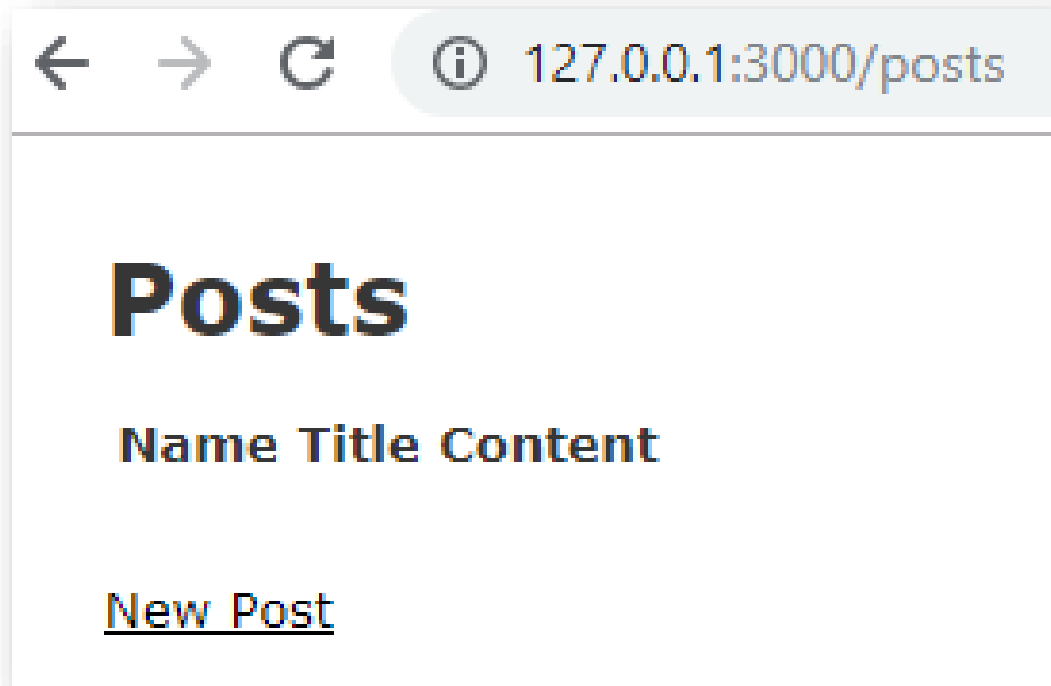
GemFile

- Temos que adicionar:
 - `gem 'coffee-script-source', '1.8.0'`
- Gems são bibliotecas existentes que precisam ser instaladas
- Usamos o bundle

A terminal window with a dark background. The title bar says "Run anything:" in orange. Below it, a command is entered: `bundle update coffee-script-source`. The command is preceded by a small green play button icon. A settings gear icon is in the top right corner.

```
Run anything: bundle update coffee-script-source
```

Rodando o servidor novamente




Como saber quais os endereços disponíveis?

Routes

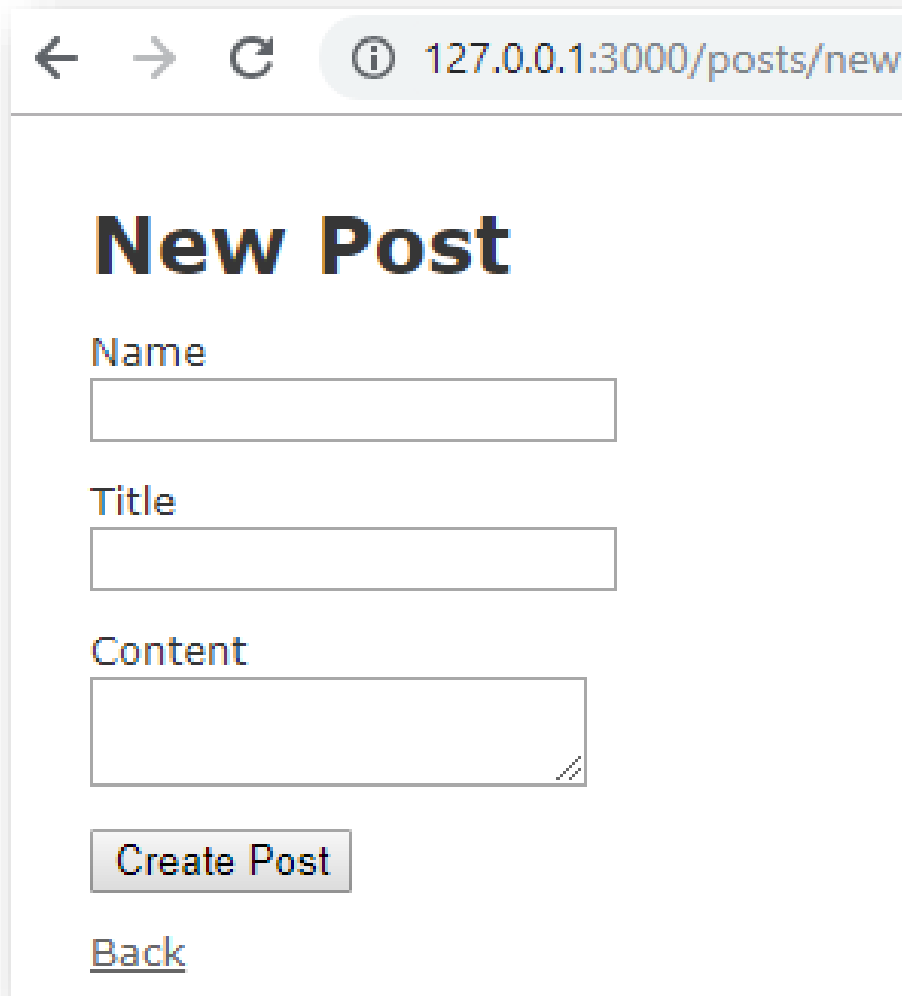
Run anything:



 rake routes

Prefix	Verb	URI Pattern	Controller#Action
posts	GET	/posts(.:format)	posts#index
	POST	/posts(.:format)	posts#create
new_post	GET	/posts/new(.:format)	posts#new
edit_post	GET	/posts/:id/edit(.:format)	posts#edit
post	GET	/posts/:id(.:format)	posts#show
	PATCH	/posts/:id(.:format)	posts#update
	PUT	/posts/:id(.:format)	posts#update
	DELETE	/posts/:id(.:format)	posts#destroy

Acessando página de criar post



A screenshot of a web browser window. The address bar shows the URL `127.0.0.1:3000/posts/new`. The page title is **New Post**. Below the title, there are three input fields: 'Name', 'Title', and 'Content'. The 'Content' field is a larger text area with a small icon in the bottom right corner. Below the input fields, there is a button labeled 'Create Post' and a link labeled Back.

← → ↻ ⓘ 127.0.0.1:3000/posts/new

New Post

Name

Title

Content

Create Post

[Back](#)

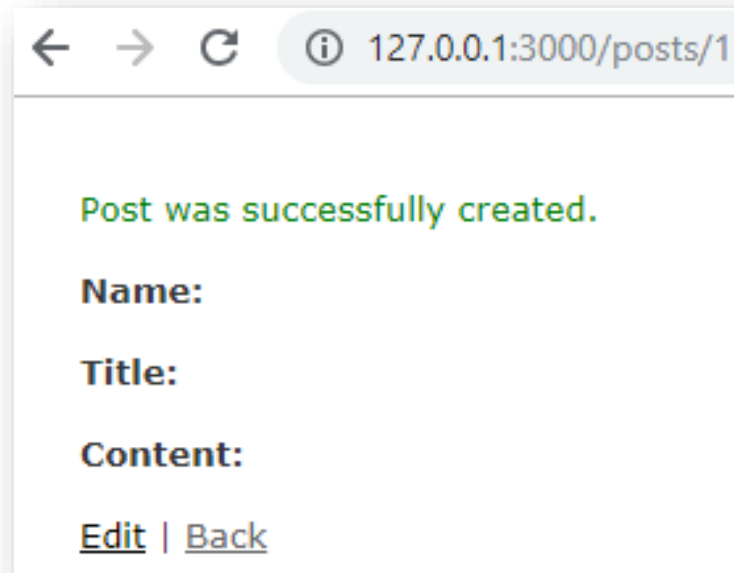
Scaffold cria vários arquivos

File	Purpose
db/migrate/20100207214725_create_posts.rb	Migration to create the posts table in your database (your name will include a different timestamp)
app/models/post.rb	The Post model
test/unit/post_test.rb	Unit testing harness for the posts model
test/fixtures/posts.yml	Sample posts for use in testing
config/routes.rb	Edited to include routing information for posts
app/controllers/posts_controller.rb	The Posts controller
app/views/posts/index.html.erb	A view to display an index of all posts
app/views/posts/edit.html.erb	A view to edit an existing post
app/views/posts/show.html.erb	A view to display a single post

Scaffold cria vários arquivos

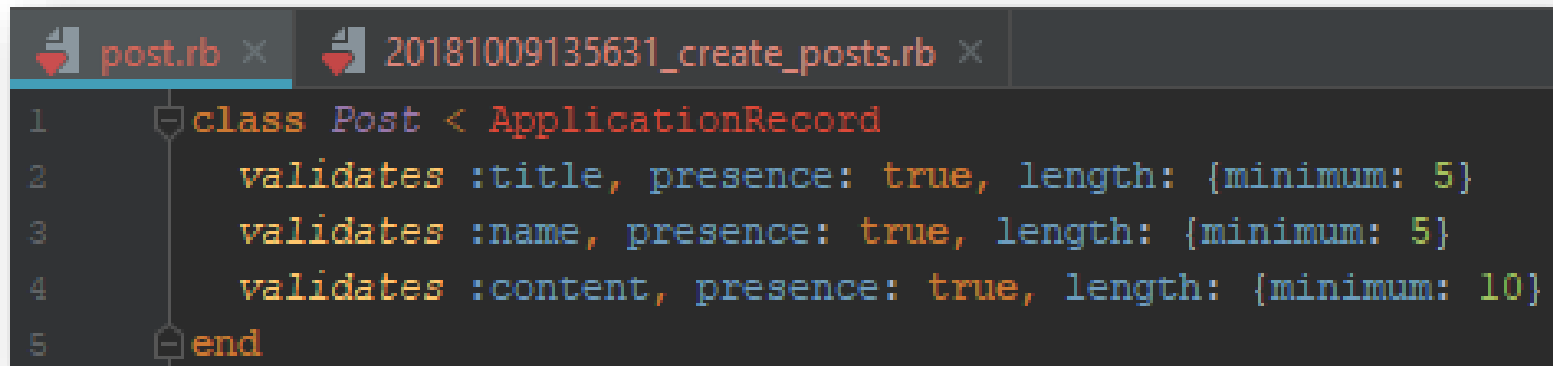
<code>app/views/posts/_form.html.erb</code>	A partial to control the overall look and feel of the form used in edit and new views
<code>test/functional/posts_controller_test.rb</code>	Functional testing harness for the posts controller
<code>app/helpers/posts_helper.rb</code>	Helper functions to be used from the post views
<code>test/unit/helpers/posts_helper_test.rb</code>	Unit testing harness for the posts helper
<code>app/assets/javascripts/posts.js.coffee</code>	CoffeeScript for the posts controller
<code>app/assets/stylesheets/posts.css.scss</code>	Cascading style sheet for the posts controller
<code>app/assets/stylesheets/scaffolds.css.scss</code>	Cascading style sheet to make the scaffolded views look better

Faz sentido criar um Post sem título?



Precisamos definir regras!

Validation



The screenshot shows a code editor with two tabs: 'post.rb' and '20181009135631_create_posts.rb'. The 'post.rb' tab is active and displays the following Ruby code:

```
1 class Post < ApplicationRecord
2   validates :title, presence: true, length: {minimum: 5}
3   validates :name, presence: true, length: {minimum: 5}
4   validates :content, presence: true, length: {minimum: 10}
5 end
```

New Post

6 errors prohibited this post from being saved:

- Title can't be blank
- Title is too short (minimum is 5 characters)
- Name can't be blank
- Name is too short (minimum is 5 characters)
- Content can't be blank
- Content is too short (minimum is 10 characters)

Name

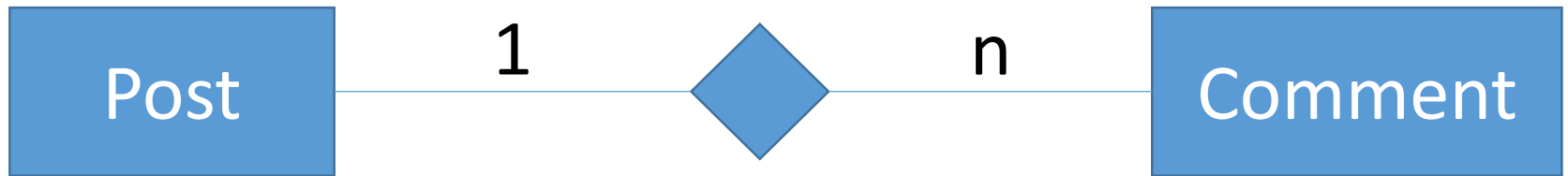
Title

Content

Create Post

[Back](#)

Relacionamentos



Run anything:



```
rails generate scaffold Comment author:string content:text
```

```
rails generate scaffold Comment author:string content:text post:references
```

Run anything:



```
rake db:migrate
```

Associations

```
post.rb x
1 class Post < ApplicationRecord
2   has_many :comments, dependent: :destroy
3   validates :title, presence: true, length: {minimum: 5}
4   validates :name, presence: true, length: {minimum: 5}
5   validates :content, presence: true, length: {minimum: 10}
6 end
```

```
post.rb x comment.rb x
1 class Comment < ApplicationRecord
2   belongs_to :post
3 end
```

Associations

```
v.html.erb x 20181009182015_create_comments.rb x post
class CreateComments < ActiveRecord::Migration[5.1]
  def change
    create_table :comments do |t|
      t.string :author
      t.text :content
      t.references :post, foreign_key: true

      t.timestamps
    end
  end
end
```


Route para comentários

```
routes.rb x
1  Rails.application.routes.draw do
2    resources :comments
3    resources :posts
4  end
```

```
routes.rb x
1  Rails.application.routes.draw do
2    resources :posts do
3      resources :comments
4    end
5  end
```

Rake routes

	Prefix	Verb	URI Pattern	Controller#Action
post_comments		GET	/posts/:post_id/comments(:format)	comments#index
		POST	/posts/:post_id/comments(:format)	comments#create
new_post_comment		GET	/posts/:post_id/comments/new(:format)	comments#new
edit_post_comment		GET	/posts/:post_id/comments/:id/edit(:format)	comments#edit
post_comment		GET	/posts/:post_id/comments/:id(:format)	comments#show
		PATCH	/posts/:post_id/comments/:id(:format)	comments#update
		PUT	/posts/:post_id/comments/:id(:format)	comments#update
		DELETE	/posts/:post_id/comments/:id(:format)	comments#destroy
posts		GET	/posts(:format)	posts#index
		POST	/posts(:format)	posts#create
new_post		GET	/posts/new(:format)	posts#new
edit_post		GET	/posts/:id/edit(:format)	posts#edit
post		GET	/posts/:id(:format)	posts#show
		PATCH	/posts/:id(:format)	posts#update
		PUT	/posts/:id(:format)	posts#update
		DELETE	/posts/:id(:format)	posts#destroy

Mostrando os comentários

```
<h2>Comments</h2>
<% @post.comments.each do |comment| %>
  <p>
    <strong>Committer:</strong>
    <%= comment.author %>
  </p>

  <p>
    <strong>Comment:</strong>
    <%= comment.content %>
  </p>
<% end %>
```

Criando novos comentários

```
<h2>Add a comment:</h2>
<%= form_for([@post, @post.comments.build]) do |f| %>
  <div class="field">
    <%= f.label :author %><br />
    <%= f.text_field :author %>
  </div>
  <div class="field">
    <%= f.label :content %><br />
    <%= f.text_area :content %>
  </div>
  <div class="actions">
    <%= f.submit %>
  </div>
<% end %>
```

Salvando os comentários

```
def create
  @post = Post.find(params[:post_id])
  @comment = @post.comments.create(comment_params)
  redirect_to post_path(@post)
end
```

Name: Testando um post

Title: o titulo eu vou escolher

Content: Ainda vou escrever o conteudo

Comments

Author: Rodrigo

Comment: ainda vou decidir

Author: Rodrigo

Comment: Ainda vou decidir

Add a comment:

Author

Content

Create Comment

[Edit](#) | [Back](#)



Referências



https://guides.rubyonrails.org/getting_started.html

Capítulos 4 e 5