

# Ruby On Rails: Laboratório 02

Laboratório Engenharia de Software



**PUC Minas**  
**Poços de Caldas**

Luiz Alberto Ferreira Gomes

Curso de Ciência da Computação

16 de abril de 2020

# Agenda

---

- 1 Ruby on Rails
- 2 Metodologia de Trabalho
- 3 Esqueleto da Aplicação
- 4 Primeira Aplicação
- 5 Versionando a Primeira Aplicação
- 6 Para Saber Mais

# Rails

---

Rails é um **framework** para construção de **aplicações web** baseado na **linguagem Ruby**.

# Rails

---

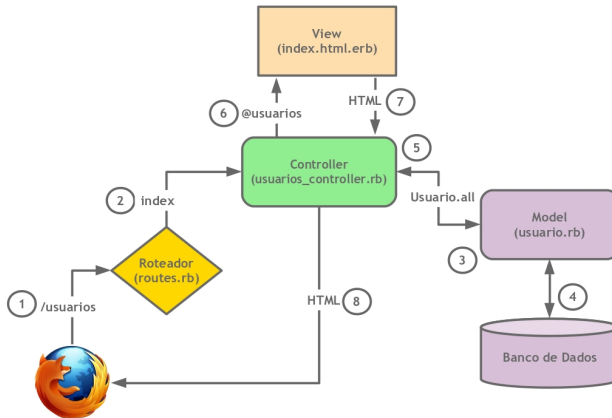
- Uma **gem** (pacote) escrita em Ruby
- 100% **open-source**, MIT License
- Fornece **geradores** de código e scripts de **automação** de testes
- Ferramentas adicionais são fornecidas no ecossistema Rails:

# Ferramentas Adicionais

---

- **Rake** - utilitário similar ao **make do Unix** para criar e migrar bancos de dados, limpar sessões de uma Web app
- **Puma** - servidor web de desenvolvimento para execução de aplicações Rails
- **SQLite** - um servidor de banco de dados simples pré-instalado como o Rails
- **Rack Middleware** - interface padronizado para interação entre um servidor web e uma Web App

# Model-View-Controller



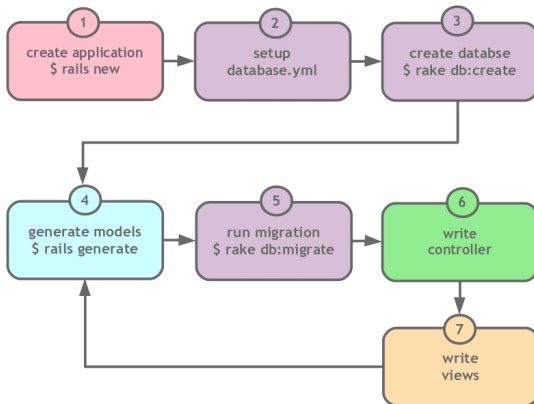
# Agenda

---

- 1 Ruby on Rails
- 2 Metodologia de Trabalho**
- 3 Esqueleto da Aplicação
- 4 Primeira Aplicação
- 5 Versionando a Primeira Aplicação
- 6 Para Saber Mais

# Metodologia de Trabalho

---





# Agenda

---

- 1 Ruby on Rails
- 2 Metodologia de Trabalho
- 3 Esqueleto da Aplicação**
- 4 Primeira Aplicação
- 5 Versionando a Primeira Aplicação
- 6 Para Saber Mais

# Estrutura de uma Aplicação Rails (1)

---

File/Folder	Purpose
app/	Contains the controllers, models, views, helpers, mailers, channels, jobs, and assets for your application. You'll focus on this folder for the remainder of this guide.
bin/	Contains the rails script that starts your app and can contain other scripts you use to setup, update, deploy, or run your application.
config/	Configure your application's routes, database, and more. This is covered in more detail in <a href="#">Configuring Rails Applications</a> .
config.ru	Rack configuration for Rack based servers used to start the application. For more information about Rack, see the <a href="#">Rack website</a> .
db/	Contains your current database schema, as well as the database migrations.
Gemfile Gemfile.lock	These files allow you to specify what gem dependencies are needed for your Rails application. These files are used by the Bundler gem. For more information about Bundler, see the <a href="#">Bundler website</a> .

# Estrutura de uma Aplicação Rails (2)

---

lib/	Extended modules for your application.
log/	Application log files.
package.json	This file allows you to specify what npm dependencies are needed for your Rails application. This file is used by Yarn. For more information about Yarn, see the <a href="#">Yarn website</a> .
public/	The only folder seen by the world as-is. Contains static files and compiled assets.
Rakefile	This file locates and loads tasks that can be run from the command line. The task definitions are defined throughout the components of Rails. Rather than changing Rakefile, you should add your own tasks by adding files to the lib/tasks directory of your application.
README.md	This is a brief instruction manual for your application. You should edit this file to tell others what your application does, how to set it up, and so on.

# Estrutura de uma Aplicação Rails (3)

---

storage/	Active Storage files for Disk Service. This is covered in <a href="#">Active Storage Overview</a> .
test/	Unit tests, fixtures, and other test apparatus. These are covered in <a href="#">Testing Rails Applications</a> .
tmp/	Temporary files (like cache and pid files).
vendor/	A place for all third-party code. In a typical Rails application this includes vendored gems.
.gitignore	This file tells git which files (or patterns) it should ignore. See <a href="#">GitHub - Ignoring files</a> for more info about ignoring files.
.ruby-version	This file contains the default Ruby version.

# Agenda

---

- 1 Ruby on Rails
- 2 Metodologia de Trabalho
- 3 Esqueleto da Aplicação
- 4 Primeira Aplicação**
- 5 Versionando a Primeira Aplicação
- 6 Para Saber Mais

# Prática de Laboratório 1 (1)

---

1. Inicie uma janela de terminal e digite no prompt:

```
$ rails new blog
```

2. Mude para o diretório da aplicação (RAILS.root)

```
$ cd blog
```

3. Execute o servidor web embutido:

```
$ rails server
```

4. Abra uma janela do navegador e digite:

```
http://localhost:3000
```

## Prática de Laboratório 1 (2)

---

5. Verifique o conteúdo do arquivo de configuração `database.yml`:

```
$ cat config/database.yml
```

6. Crie o banco de dados de desenvolvimento e testes:

```
$ rake db:create
```

7. Crie o modelo Post:

```
$ rails g model Post title:string body:text
```

8. Implemente modelo Post no banco de dados com *migrations*:

```
$ rake db:migrate
```

# Prática de Laboratório 1 (3)

---

## 9. Crie o controlador PostsController:

```
$ rails g controller Posts
```

## 10. Modifique o arquivo config/routes.rb para acrescentar as rotas para o recurso posts:

```
1 Rails.application.routes.draw do
2   resources :posts
3 end
```

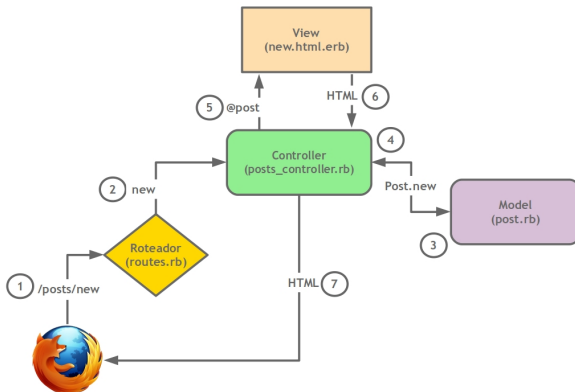
## 11. Execute o comando rake para visualizar as rotas para os posts:

```
$ rake routes
```

## 12. Reinicie o servidor web e acesse a url <http://localhost:3000/posts/new>. Veja o erro que ocorreu.



# Ação: New (1)



## Ação: New (2)

---

- Um novo objeto `@post` da classe `Post` é instanciado
- Procura pela visão `new.html.erb` para renderizar a resposta

Listing 1: `app/controllers/posts_controller.rb`

```
1 Class PostsController < ApplicationController
2   def new
3     @post = Post.new
4   end
5 end
```

# Visão: New (1)

---

- Reinicie o servidor web e acesse a url `<http:\localhost:3000/posts/new>`. Veja o erro que ocorreu.
- Implemente a visão `new.html.erb`:

## Listing 2: views/posts/new.html.erb

```
1 <h1>Novo Post</h1>
2 <%= form_with scope: :post, url: posts_path,
3   local: true do |form| %>
4   <p>
5     <%= form.label :title %><br>
6     <%= form.text_field :title %>
7   </p>
8
9   <p>
10    <%= form.label :body %><br>
```

## Visão: New (2)

---

```
11      <%= form.text_area :body %>
12    </p>
13
14    <p>
15      <%= form.submit %>
16    </p>
17  <% end %>
```

## Prática de 2 (1)

---

1. Implemente a ação e a visão New na aplicação blog.

# Agenda

---

- 1 Ruby on Rails
- 2 Metodologia de Trabalho
- 3 Esqueleto da Aplicação
- 4 Primeira Aplicação
- 5 Versionando a Primeira Aplicação**
- 6 Para Saber Mais

# Controle Automatizado de Versão (1)

---

- GitHub
- git
- git config
- git init
- git add
- git commit
- git remote
- git push

# GitHub

---



Figura: [www.github.com](https://www.github.com)



# git

---



Figura: <https://git-scm.com/>

# git + GitHub

---



Figura: <https://git-scm.com/>

# Agenda

---

- 1 Ruby on Rails
- 2 Metodologia de Trabalho
- 3 Esqueleto da Aplicação
- 4 Primeira Aplicação
- 5 Versionando a Primeira Aplicação
- 6 Para Saber Mais**

# Para Saber Mais

---

- `<https://www.ruby-lang.org/en/>`
  - referência oficial da linguagem Ruby onde a toda a sua documentação está disponível para ser consultada.
- `<http://rubyonrails.org/>`
  - referência oficial do framework Rails onde a toda a sua documentação está disponível para ser consultada.
- `<http://www.codecademy.com/pt/tracks/ruby>`
  - curso iterativo em português sobre a linguagem Ruby.
- `<https://gorails.com/setup/ubuntu/16.04>`
  - guia para instalação do Ruby on Rails no Ubuntu e no Mac OSX.