

Lista de exercícios semana 3

Controller

Fala meus compatriotas, como vão indo? Espero que estejam todos bem. Para começar a semana bem, saibam que a lista dessa semana é bem tranquila. O Prazo de entrega é de duas semana, mas lembrando, sei das dificuldades para manter o processo trainee e as aulas da UnB, não se preocupem muito com prazo, o importante é entregar as atividades, já que está é a melhor maneira de se aprender a programar.

O conteúdo desta semana é Controller, então para essa lista, gostaria que vocês fizessem um programinha em Rails com 2 Controller

A primeira para Escritores; A segunda para Livros

- Os escritores terão que ter o campo Nome, Ano de nascimento, país de origem.
- Já os livros, terão que ter Nome, Ano da publicação, Escritor.

As controller devem seguir o modelo CRUD (Create, Read, Update, Delete) que foi os métodos criado em aula.

São necessárias as Views (index, show, new, edit).html.erb

Seria legal se o nome do escritor fosse obrigatório, assim como o nome do livro e o autor dele.

Ponto extra

Utilizar `Scope 'livros'` do `Em rotas`

Talvez ajude (alguns exemplos que podem ser útil)

```
# na index, eu pegaria todos os livros (algo como @books = Book.all)
# Seria legal eu utilizar uma função do ruby para imprimir isso na página inicial

<% @books.each do |livro| %>
  # dados do livro
<% end %>
```

Ruby ▾

```
# não se esqueçam do formulario quando for envia algo para o banco

<%= form_for @book do |livro| %>
  # dados para o usuario digitar
<%= livro.submit %>

# e lembrem que seria legal pegar os parametros de um método privado
```

Ruby ▾

```
# Quando eu for criar ou editar algum dados seira bom eu verificar se existe erro
# ao invéz de utilizar .create eu poderia .new e o .save
# posso utilizar o flash para notificar se deu certo ou não

@book = Book.new(book_params)
if (book.save)
  # se salvo corretamente
else
  # caso não salvou corretamente

# Ou posso utilizar outra maneira

@book = Book.new(book_params)
begin
  @movie.save!
  # se salvo corretamente
rescue => exception
  # caso não salvou corretamente, o programa lança exceção
ensure
  # independente do que aconteceu eu vou para essa linha
```

Ruby ▾