PUTS_DEV FRAMEWORK

Aprenda a desenvolver o seu próprio framework WEB

html

RUBY

Adyson Lima

Sumário

Capítulo 1

Qual o seu propósito ao ler este mini e-book?

Caro leitor, sempre que escrevo um tutorial ou e-book, eu faço um pequeno incentivo para a reflexão do leitor. E neste mini e-book, se me permite, farei o mesmo. Assim…

Qual o seu propósito ao ler este material?

Talvez seja apenas passar o tempo. Talvez aprender algo novo. Ou talvez seu propósito seja algo mais profundo como:

“ *Meu propósito é aprender com este material e usar o conhecimento para somar aos conhecimentos que já possuo e assim, poder solucionar problemas reais com maior qualidade e profissionalismo. E além disso, com os ganhos financeiros obtidos de minha atividade profissional, ajudar pessoas em situação de morador de rua. Completando assim meu ciclo de realização profissional e acrescentando minha parcela de contribuição para a sociedade.* ”

Ok, isso foi tocante, muito bom. Mas brincadeiras a parte, **tenha um propósito**, que motive você a **agir** e que **justifique** seu empenho em aprender. Isso é muito importante para seus futuros resultados.

Capítulo 2

Sobre o potencial do brasileiro.

Caro leitor, é notório o fato de que o povo brasileiro é lutador, persistente e criativo. Sim, o brasileiro possui muito potencial. Veja exemplos como:

- Zilda Arns e seu trabalho de cuidado com a saúde infantil.

- Alberto Santos Dumont e seu 14-bis.

- Hortência Marcari e sua trajetória de sucesso no basquetebol.

- Oscar Niemeyer e o projeto arquitetônico de Brasília.

Esses são alguns exemplos de brasileiros, como você caro leitor. E na área de tecnologia, temos grandes potenciais? Sim, com certeza, **você é um desses** potenciais.

E falando ainda em potenciais na área de tecnologia no Brasil, vou citar um desenvolvedor Ruby, entre muitos outros grandes talentos de T.I. que temos no Brasil, o Rodrigo Pimentel Satyro, o Xita, do canal puts_dev do Youtube. Ele é o desenvolvedor do Web framework puts_dev.

Em algumas horas de live, e várias de pesquisas e testes, ele desenvolveu um WEB framework, baseado principalmente em Ruby, num prazo de menos de 30 dias. Mostrando assim um pouco do potencial do desenvolvedor de software brasileiro.

E é baseado neste WEB framework, o puts_dev framework que será escrito este mini e-book.

Então, existe muito potencial no brasileiro, e isso é muito bom, somos brasileiros.

Capítulo 3

Uma excelente oportunidade de aprender OU mais um framework?

Este mini e-book, trata de um framework em desenvolvimento, atualmente apenas seu criador, o Xita, desenvolve ele. Observe:

1° - O processo de criação deste framework Web, do zero até o estado atual, está disponível no Youtube.

2° - O conteúdo em vídeo e este mini e-book são 100% gratuitos.

3° - Você pode aumentar sua habilidade de programação em Ruby codificando assistindo os vídeos.

4° - Todo o código fonte do framework está no Github.

Então, possivelmente o puts_dev framework é uma oportunidade excelente de melhorar suas habilidades de programação. Pense a respeito.

- Você teve a oportunidade de ver o desenvolvimento de algum framework Web, acompanhando do zero o processo?

Caro leitor, você deseja melhorar suas habilidades como programador Ruby? Deseja melhorar sua habilidade em desenvolvimento Web? Deseja conhecer detalhes de como produzir um framework? Deseja aprender a desenvolver ferramentas em Ruby?

Bem, essa pode talvez ser a oportunidade. E eu lhe desejo sucesso na sua jornada caro leitor e desenvolvedor de software brasileiro e faço uma pequena sugestão.

Tente **escrever o código** além de assistir os vídeos no Youtube, isso pode trazer resultados mais interessantes ;)

Capítulo 4

Como surgiu o puts_dev framework?

De onde nasceu a ideia de desenvolver um WEB framework usando Ruby?

Bem, o Xita é desenvolvedor Rails, já usa Ruby. Mas um dia ele viu um vídeo no Youtube e decidiu criar algo a partir daí.

E o processo se deu inicialmente, na semente de código, na classe Socket do Ruby, que traz na sua documentação o pequeno exemplo a seguir. Que cria um mini servidor.

```
require 'socket'

server = TCPServer.new 2000 # Server bound to port 2000

loop do
  client = server.accept    # Wait for a client to connect
  client.puts "Hello !"
  client.puts "Time is #{Time.now}"
  client.close
end
```

A partir daí, existe uma série de vídeos que mostram o processo de desenvolvimento. Mas chega de espera, vamos usar um pouco o puts_dev framework.

Capítulo 5

Requisitos e comandos

Caro leitor, é preciso dizer que o puts_dev framework é uma ferramenta em desenvolvimento, e não tem condições e nem o interesse de substituir qualquer ferramenta conhecida do mercado. Ele possui limitações sim, mas está em desenvolvimento, pode receber melhorias.

Mas pode também, já ser usado para desenvolver aplicações simples WEB, você porá em produção e criará um novo mercado com ele? Receio que não seja possível neste momento. Mas, quem sabe no futuro?

Dito isso, vamos a alguns requisitos para usar o puts_dev framework.

1° - Ter instalada as gems: **pg**, **yaml**, **date**, **uri**, **cgi**.

2° - Ter instalado o banco de dados PostgreSQL e um gerenciador como PGADMIN, ou outro de sua escolha.

3° - Ter configurado um usuário no PostgreSQL. com privilégios de administrador, ou quase administrador ;)

Tendo esses recursos, vamos para os próximos passos, download, conhecimento básico e comandos.

- Download do puts_dev framework

Baixar o puts_dev framework é muito simples, basta ir ao Github dele e fazer o download ou o clone do repositório, como você preferir. Se optar pelo download, será preciso descompactar o arquivo.

- Conhecendo o puts_dev framework

Ao descompactar a pasta, ou clonar o repositório, você irá encontrar uma estrutura de diretórios como a que segue. Uma rápida descrição da estrutura será dada.

puts_dev_framework-main

bin

config

monkey_patches

db

migrations

lib

controllers

models

queries

views

Gemfile

Readme.md

A pasta **puts_dev_framework-main**, é a pasta principal do projeto. A pasta **bin** contém os arquivos framework.rb e server.rb que possuem a raiz das funcionalidades do framework. A pasta **config/monkey_patches** contém uma pequena customização de String para o projeto.

A pasta **db/migrations** possui as migrações geradas pelo framework. E as pastas, **lib/controllers, lib/models** e **lib/views** contém a estrutura MVC do framework. A pasta **lib/queries** contém utilidades de pesquisa ao banco de dados.

Finalmente, o arquivo **Gemfile**, contém as gems necessárias ao projeto. E o arquivo README.md possui algumas informações sobre o projeto, tal como padrão.

Além disso, é importante ressaltar que, o PostgreSQL é o banco padrão usado pelo puts_dev framework. E o CDN do CSS do Bootstrap 5 já vem configurado por padrão.

Vamos a seguir, ver um pouco sobre os comandos que o framework possui.

- Comandos básicos

O puts_dev framework possui alguns comandos que ajudam na utilização, veja a lista a seguir com os comandos e a explicação do que eles fazem.

ruby ./bin/framework.rb server este comando executa o servidor e o deixa escutando na porta 3000.

ruby ./bin/framework.rb db create:db nome_do_banco este comando cria um banco para uso no projeto.

ruby ./bin/framework.rb db drop:db nome_do_banco este comando exclui um banco.

ruby ./bin/framework.rb db create:migration NomeDaMigration este comando cria uma migração, como no exemplo mostrado nos vídeos do curso, o comando pode ser usado assim:

ruby ./bin/framework.rb db create:migration CreateProductsTable, onde foi criada uma migração para um model Product.

ruby ./bin/framework.rb db migrate este comando roda a migração e configura um banco para uso no projeto.

find . -name '*.rb' | xargs wc -l este comando é um apendice que o Xita incluiu no README do projeto no Github, ele lista a quantidade de linhas do projeto ;)

Finalizada esta breve explicação de comandos, “vamos tentar colocar gasolina e dar uma volta no quarteirão”, por assim dizer.

Capítulo 6

Uma primeira aplicação.

Neste capítulo, vamos ver a criação de uma pequena aplicação, simples, mas que mostra já a funcionalidade do framework. Não nos prenderemos a padrões e regras neste momento, vamos apenas demonstrar o potencial do puts_dev framework.

Lembre de ter os requisitos já instalados em sua máquina.

Concluindo, este mini e-book, será atualizado futuramente, então, vamos aos passos que seguiremos, neste momento.

1° baixar o framework.
2° descompactar a pasta.
3° criar um banco no PostgreSQL.
4° criar uma migração.
5° editar uma migração.
6° rodar uma migração.

1° passo.

Faça o download do framework aqui. Ok, vamos ao segundo passo.

2° passo.

Descompacte a pasta baixada no seu sistema, lembre de revisar a descrição da estrutura do projeto para se familiarizar.

3° passo.

Rode o comando de geração de banco no PostgreSQL. Como mostrado a seguir. No nosso caso, criaremos o banco de nome **project**. Atente para o fato que o arquivo **puts_dev_framework-main/config/database.yml** contém as informações de acesso ao banco, como senha e password. Certifique-se de que seu banco de dados PostgreSQL e seu arquivo estão condizentes, se não estiverem, faça ajustes de acordo com sua necessidade.

$ ruby ./bin/framework.rb db create:db project

4° passo.

Com o banco criado, vamos criar uma migração. Lembrando que o comando de criação de migração cria um modelo que você pode editar segundo sua necessidade. Nós faremos uma edição na migration no próximo passo.

$ ruby ./bin/framework.rb db create:migration CreateProductsTable

5° passo.

Com nosso banco criado e nosso modelo de migração criado, vamos editar o modelo de migração para atender nossa necessidade. Para isso, abra o arquivo de migração que está em **puts_dev_framework-main/db/migrations** e deixe seu conteúdo como a seguir.

```
class CreateProductsTable
  def self.migrate

    sql = <<~SQL
      CREATE TABLE products (
        id SERIAL PRIMARY KEY,
        name varchar(255) NOT NULL,
        price decimal NOT NULL,
        created_at date NOT NULL,
        updated_at date NOT NULL
      )
    SQL


    connection = Database.connection
    connection.exec(sql)
    connection&.close
  end
end

CreateProductsTable.migrate
```

Observe os campos **name** e **price** que serão usados na nossa aplicação, bem como o nome da tabela, **products**, ambos em vermelho. E veja que o arquivo inclui uma seção de SQL, que começa em "**sql =** " que pode ser editada de acordo com a necessidade do projeto. Por exemplo, se quiséssemos um campo quantity, bastaria inserir, de acordo com o modelo. Resta agora rodar a migração, o que faremos a seguir.

6° passo.

Agora, vamos rodar a nossa migração, como o comando a seguir.

```
$ ruby ./bin/framework.rb db migrate
```

Pronto, temos o banco de dados configurado para uso. Agora, vamos rodar a aplicação com o comando a seguir e abrir a aplicação na página de criação de produtos no endereço **localhost:3000/products/new**.

```
$ ruby ./bin/framework.rb server
```

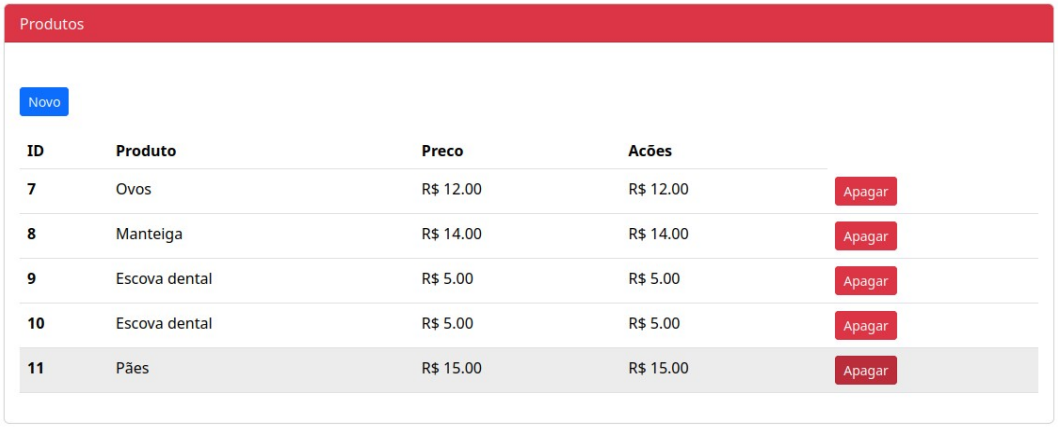
A tela da aplicação será como a da imagem a seguir.

No linha de comando, observe as saídas indicando as operações feitas no PostgreSQL., como exemplo a seguir.

```
Received -> request: GET, to: /products/new, http version: HTTP/1.1
Returning -> controller: products, action: new, params: []}
Received -> request: POST, to: /products, http version: HTTP/1.1
Returning -> controller: products, action: create, params: [{:name=>"Carro"}, {:price=>"1000"}]}
```

Neste ponto, temos condições de realizar a construção de CRUD's. E agora vamos acrescentar um pouco de estilização com Bootstrap no arquivo **puts_dev_framework-main/lib/views/products/index_html.rb.** Lembrando que o CDN do Bootstrap já vem incluso no puts_dev framework por padrão, assim, você pode usar as classes CSS e fazer customizações de acordo com sua vontade.

Segue uma imagem do arquivo editado com Bootstrap.



Concluímos aqui nossa demostração de uso do puts_dev framework, lembre que há ainda limitações, mas como dito, este mini e-book e o próprio framework estão em evolução ainda. Mais informação sobre o framework virá em breve. **Os arquivos usados neste mini e-book estão no Github deste material**.

Agradecimentos

Agradeço primeiramente a Deus e a meus pais que sempre me ajudaram. E agradeço ao Xita pela oportunidade de aprender um pouco mais e a você leitor. Muito Obrigado.

Referências

Canal puts_dev
Readme do puts_dev framework