



3. Rails Básico

- a. Instalação do rails
 - i. Gerenciador RubyGems: bibliotecas para ruby
 - ii. `gem install rails (--include-dependencies)`
`gem install sqlite3-ruby` ou `gem install mysql (--include-dependencies)`
 - iii. Usando o sqlite3 não é necessário instalar um SGBD
 1. Banco de dados padrão do rails desde a versão 2.0
 2. Bastante fácil alterar o banco de dados depois
 - iv. `rails --version` (verificar a versão)
- b. Criando a primeira aplicação
 - i. `rails horas_trabalhadas` (muito simples)
 - ii. Saída
 1. `create`
`create app/controllers`
`create app/helpers`
`create app/models`
`create app/views/layouts`
...
`create log/server.log`
`create log/production.log`
`create log/development.log`
`create log/test.log`
- c. Estrutura de diretórios de uma aplicação rails
 - i. app: código da aplicação rails
 1. controllers: controladores, tratarão as requisições HTTP
 2. helpers: repositórios de métodos, para serem usados nas views
 3. models: classes de configuração e persistência e lógica de negócios
 4. views: views e layouts da aplicação
 - ii. config: arquivos de configuração
 1. environments: configurações específicas para cada ambiente da aplicação (development, test e production)
 2. initializers: arquivos a serem executados na inicialização da aplicação
 - iii. db: não usado pelo framework, as vezes armazena o banco de dados de desenvolvimento
 1. migrations: migrações para o banco de dados
 - iv. doc: onde será gerada a documentação do código
 - v. lib: qualquer código ruby que não seja um controller, helper ou model
 1. tasks: tarefas do Rake personalizadas para a aplicação



UNIVERSIDADE FEDERAL DO PIAUÍ
CENTRO DE CIÊNCIAS DA NATUREZA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

- vi. log: depurar a aplicação depois que ela entrar em produção (logs de tudo o que acontece)
- vii. public: arquivos estáticos que devem ser publicados para o usuário. HTML simples, imagens, css, javascript
 1. images: imagens
 2. javascripts: scripts em javascript
 3. stylesheets: folhas de estilo
- viii. script: diversos scripts do rails (generate, console, dbconsole, etc)
- ix. test: testes
 1. fixtures: arquivos YAML para carregar dados de teste
 2. functional: testes de aceitação
 3. integration: testar os controladores da aplicação
 4. unit: testes para os models
- x. tmp: arquivos temporários do framework
- xi. vendor: fundamental para funcionar no servidor, com subdiretorio para plug-ins. Podes-se congelar plugins aqui
- d. Scripts padrão do rails
 - i. Por enquanto apenas os mais utilizados
 1. about: versão do rails e outras dependencias usadas pela aplicação
 2. console: console semelhante ao irb, mas com os dados da aplicação rails carregados
 3. dbconsole: abre o cliente padrão do banco de dados, suporta mysql, postgres e sqlite
 4. destroy: contrário ao generate, destroi código gerado
 5. generate: cria estruturas básicas para modelo, controlador, migration e um enviador de email, além de outros geradores padrão, que iremos vendo conforme formos utilizando
 6. plugin: instala e desinstala plugins
 7. runner: executar via console códigos ruby que acessem dados da aplicação (automatizar execução de tarefas)
 8. server: inicia o servidor da aplicação. Por enquanto usaremos somente ele, mais a frente teremos um capítulo somente sobre deployment da aplicação.
 - a. Rode ruby script/server e sua aplicação irá iniciar, podendo ser acessada por <http://localhost:3000>
 - ii. Prove diversas tarefas padrão para facilitar a vida do desenvolvedor
 1. rake --T (lista todas as tarefas disponíveis)
 - iii. Tarefas principais
- e. Antes de continuar precisamos conhecer outro comando: o rake
 - i. Instalação: gem install rake
 - ii. Prove diversas tarefas padrão para facilitar a vida do desenvolvedor
 1. rake --T (lista todas as tarefas disponíveis)
 - iii. Tarefas principais



UNIVERSIDADE FEDERAL DO PIAUÍ
CENTRO DE CIÊNCIAS DA NATUREZA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

1. rake db:charset caractere set usado pelo banco
 2. rake db:create cria o banco de dados para o ambiente atual (por padrão development)
 3. rake db:create:all cria os bancos para todos os ambientes
 4. rake db:drop apaga o banco do ambiente atual
 5. rake db:drop:all apaga todos os bancos configurados
 6. rake db:migrate aplica as migrações ao banco atual
 7. rake db:reset mesmo que executar db:drop e dp:create em sequência
 8. rake notes lista os comentários marcados como FIXME, TODO e OPTIMIZE
 9. rake notes:fixme lista apenas os FIXME
 10. rake notes:optimize lista apenas os OPTIMIZE
 11. rake notes:todo lista apenas os TODO
 12. rake routes lista todas as rotas nomeadas definidas no arquivo routes.rb
- f. Geração de código, um cadastro instantâneo
- i. Todos dizem que é muito rápido desenvolver aplicações rails
 - ii. Iremos gerar um cadastro instantaneamente, eliminando dependências de partes do sistema que ainda não estejam prontas
 1. Mesmo que boa parte não seja realmente utilizável em produção
 - iii. No console, vá ao diretório da aplicação e execute:
 1. ruby script/generate scaffold user login:string name:string email:string
rake db:create:all
rake db:migrate
ruby script/server
 2. Abra no browser <http://localhost:3000/users>
 - a. Listagem de usuários vazia
 - b. Cadastraremos usuários para ver que o sistema realmente funciona completamente
 - iv. O comando ruby script/generate scaffold gera um esqueleto de um cadastro completo e funcional. Os arquivos criados foram:
 1. db/migrate/<numeros>_create_users.rb
app/model/user.rb
app/controllers/user_controller.rb
app/views/users/index.html.erb
app/views/users/show.html.erb
app/views/users/new.html.erb
app/views/users/edit.html.erb



UNIVERSIDADE FEDERAL DO PIAUÍ
CENTRO DE CIÊNCIAS DA NATUREZA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

```
app/views/layouts/users.html.erb  
public/stylesheets/scaffold.css  
app/helpers/users_helper.rb  
test/unit/user_test.rb  
test/fixtures/users.yml  
test/functional/users_controller_test.rb  
config/routes.rb
```

- v. `config/routes` não foi realmente criado, apenas alterado. Veremos mais a frente essas alterações
- g. A primeira Migration
- h. O primeiro model
- i. O primeiro controlador
- j. Métodos básicos do ActiveRecord
- k. Recebendo parâmetros nos controladores
- l. Respondendo a requisições
- m. As primeiras views
- n. Precisando de ajuda para limpar o código das views?
- o. Configuração de rotas, um nome bonito para URLs
- p. Continuando o desenvolvimento
- q. Conferindo as migrations geradas
- r. Escrevendo modelos
- s. Gerando todo o código
- t. Um pouco de segurança na aplicação
- u. Um layout menos confuso para a aplicação
- v. Associando usuários a projetos
- w. Adicionando tipos de tarefas a um projeto
- x. Cadastro das horas trabalhadas
- y. Um relatório para a aplicação
- z. Limpando um pouco o código e se livrando um pouco do “inglês”