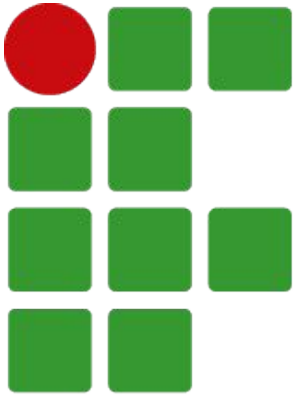


INSTITUTO FEDERAL
Paraná
Campus Paranaguá

Ruby on Rails

Desenvolvimento Web III

Prof. Diego Stiehl

INSTITUTO FEDERAL

Ruby on Rails

- Framework **web** que utiliza a linguagem Ruby
  - Formado por diversas Gems (bibliotecas)
  - Projetado para tornar o desenvolvimento web mais fácil e intuitivo
- Modelo MVC bastante puro
- CoC: Convenção sobre Configuração
  - Proporciona menos código e mais recursos
- Programação orientada a diversão do programador

INSTITUTO FEDERAL

Ruby on Rails - Filosofias

- CoC (Convention over configuration)
  - Em vez de configurar um conjunto de arquivos, adota-se uma convenção e apenas mudar o que for necessário
- DRY (Don't Repeat Yourself)
  - Nunca fazer mais de uma vez o que for necessário (como checar uma regra de negócio)
- Automação de tarefas repetitivas

INSTITUTO FEDERAL

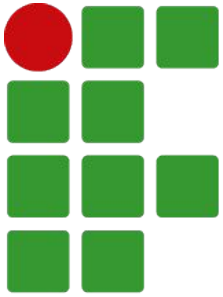
Rails - Outras Características

- Fortemente conectado a um banco de dados
  - A forma como estruturaremos as entidades depende diretamente da estrutura do BD
    - E vice-versa
- REST - Representational State Transfer
  - As “chamadas” são baseadas fortemente no que o protocolo HTTP oferece nativamente
    - URL “bonitas” e facilidade de criação de APIs

INSTITUTO FEDERAL

Tudo em Inglês

- O Rails abstrai e “adivinha” muitas coisas nós
- Para que isto funcione da forma ideal:
  - Desenvolver TOTALMENTE EM INGLÊS
    - Nomes de entidades, atributos, variáveis, arquivos, configurações, …



INSTITUTO FEDERAL
Paraná
Campus Paranaguá

Ruby on Rails

Configurações e Primeiro Projeto

INSTITUTO FEDERAL

Verificando Versões

- Vamos tentar instalar as coisas
- Vendo se já temos o Ruby e o Ruby on Rails
  - ruby -v
    - ruby 2.6.0 (no meu caso)
  - rails -v
    - Rails 5.2.3 (no meu caso)
- OK! Mas estas não são as mais recentes.
  - Última versão estável Ruby: 2.6.4
  - Última versão estável Ruby on Rails: 6.0.0

INSTITUTO FEDERAL

RVM

- RVM (Ruby Version Manager)
  - Ferramenta para automatização e gerência de instalações de diferentes versões do Ruby
- Instalação:
  - https://rvm.io/rvm/install
- Executar:
  - rvm install ruby-2.5.0
  - rvm use ruby-2.5.0
  - ruby -v

INSTITUTO FEDERAL

rbenv

- Concorrente do RVM. Eu prefiro ele.
  - Mais ferramentas para usar em produção
- Instalação:
  - https://github.com/rbenv/rbenv#basic-github-checkout
- Plugin com as versões:
  - https://github.com/rbenv/ruby-build
- Executar
  - rbenv install 2.6.0
  - rbenv global 2.6.0

INSTITUTO FEDERAL

Atualizando Rails

- O Rails é uma Gem (na verdade, são várias)
- Instalação
  - gem install rails
    - Instala a versão mais recente no RubyGems
- Quer garantir uma versão específica?
  - gem install rails -v 5.2.3

INSTITUTO FEDERAL

Criação de Projeto

- rails new <nome>
  - Cria (ou usa) diretório com o nome informado
  - Cria estrutura básica de projeto Rails
  - Cria arquivos de configuração
  - Baixa e instala dependências (Bundler)

INSTITUTO FEDERAL

Banco de Dados

- Uma aplicação Rails sempre estará vinculada a um SGBD
  - Arquivo de configuração: config/database.yml
- Banco de dados padrão:
  - sqlite3
- Se desejar criar usando outro sistema:
  - rails new <nome> -d <bd>
    - Opções:
      - **mysql**/**postgresql**/**sqlite3**/oracle/frontbase/ibm_db/sqlserver/jdbcmysql/jdbcsqlite3/jdbcpostgresql/jdbc

INSTITUTO FEDERAL

Estrutura do Projeto

- DIRETÓRIO RAIZ
  - app
    - assets
    - controllers
    - helpers
    - models
    - views
  - bin
  - config
  - Gemfile
  - db
    - migrate
  - lib
  - public
  - test
  - vendor

INSTITUTO FEDERAL

Executar a Aplicação

- Para executar (startar) a aplicação, em modo de desenvolvimento:
  - rails server
- Forma curta:
  - rails s ← "s" de server
- Podemos setar IP e porta
  - rails s -b 0.0.0.0 -p 3000
- Ficará acessível em:
  - http://ipOUlocalhost:porta

Scaffold

- Tradução: Andaime
  - Algo que nos ajuda a construir algo
- O comando scaffold permite a criação de CRUDs simples
  - Mas completos
  - Create, Retrieve, Update e Delete
    - Já conectado ao BD

INSTITUTO FEDERAL

Scaffold

- Comando
  - rails generate scaffold <Entidade> <atributo1>:<tipo1> <atributo2>:<tipo2>
- Forma curta:
  - rails g … ← “g” de generate
- Exemplo
  - rails generate scaffold Person name:string age:integer active:boolean salary:float

Singular

INSTITUTO FEDERAL

Scaffold

- Gera (ver no projeto):
  - Migration
  - Model
  - Controller
    - Ações (CRUD)
  - Conjunto de Views padrão (CRUD)
  - Rotas (URLs)
  - Helpers
  - Assets (JS/CSS)
  - Testes

INSTITUTO FEDERAL

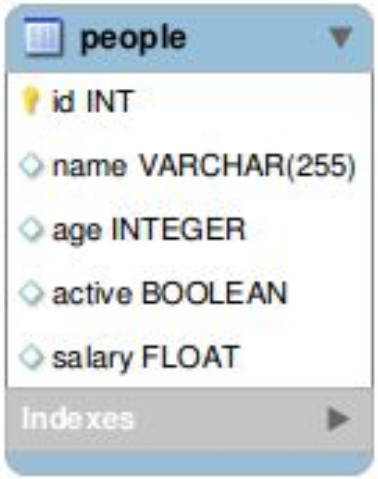
BD e Migrações

- O Scaffold gera uma versão do script do BD
  - Mas não cria ou altera um banco de dados real em si
- Para criar e atualizar um BD, precisamos rodar:
  - rails db:create
    - Cria o banco de dados (ver config/database.yml)
  - rails db:migrate
    - Executa todas as migrações de BD não executadas
      - DDL - Irá criar as tabelas, campos, …

INSTITUTO FEDERAL

No Banco de Dados

- Após a migração
  - A tabela people foi criada no banco de dados:



INSTITUTO FEDERAL

Acessando

- Os recursos criados pelo scaffold ficam disponíveis no servidor sob uma rota:
  - http://servidor/<nome_entidade_no_plural>
- Exemplo:
  - http://localhost:3000**/people**
    - Exemplo local
  - https://www.meusite.com**/people**
    - Exemplo em produção

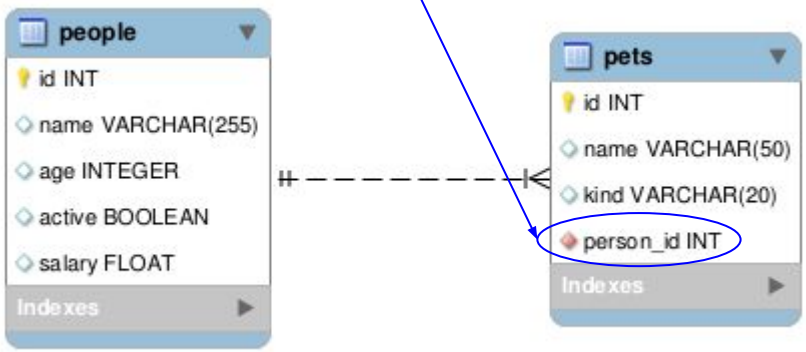
Relacionamentos

- Uma pessoa pode ter alguns pets
- Para criar o CRUD dos pets:
  - rails g scaffold Pet name kind person:references

Quando não informados o tipo, o Rails assume como :string name:string kind:string

Rails sabe automaticamente que deve criar uma relação com o model Person (tabela people)

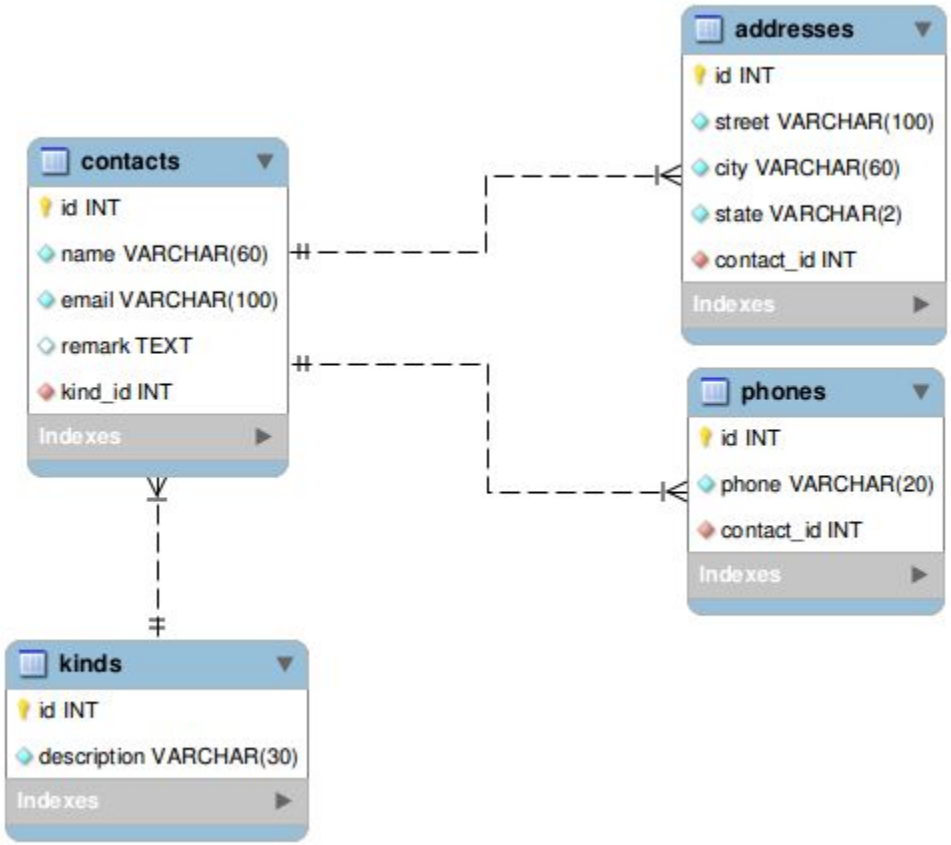
- Não esquecer de rodar:
  - rails db:migrate

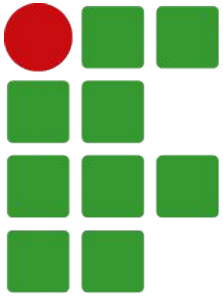


INSTITUTO FEDERAL

Prática

- Novo Projeto
  - Criar o seguinte modelo (com scaffold):





INSTITUTO FEDERAL
Paraná
Campus Paranaguá

Projeto Agenda de Contatos - Scaffold

Explorando os Recursos Criados

INSTITUTO FEDERAL

Migrações

- Migrações são versões do BD do sistema
- Só “valem” após o rails db:migrate
- O rails db:migrate executa somente as versões necessárias (verificação de data)

```
diego@diego-avell ~/projetos/rails/agenda $ ls -la db/migrate/
total 24
drwxr-xr-x 2 diego diego 4096 Set 10 20:48 .
drwxr-xr-x 3 diego diego 4096 Set 10 20:48 ..
-rw-r--r-- 1 diego diego  159 Set 10 20:45 20170910234549_create_kinds.rb
-rw-r--r-- 1 diego diego  245 Set 10 20:46 20170910234643_create_contacts.rb
-rw-r--r-- 1 diego diego  252 Set 10 20:47 20170910234724_create_addresses.rb
-rw-r--r-- 1 diego diego  203 Set 10 20:48 20170910234809_create_phones.rb
```

[ano][mês][dia][hora][minuto][segundo]

Objetivo da migration
Nome derivado do Scaffold

INSTITUTO FEDERAL

Migrações

- Verificar o banco de dados

```
diego@diego-avell ~/projetos/rails/agenda $ sqlite3 db/development.sqlite3
SQLite version 3.11.0 2016-02-15 17:29:24
Enter ".help" for usage hints.
sqlite> .tables
addresses             contacts              phones
ar_internal_metadata  kinds                 schema_migrations
sqlite> select * from schema_migrations;
20170910234549
20170910234643
20170910234724
20170910234809
sqlite>
```

Migrations já executadas (timestamp)

Estrutura de uma Migração

É criada uma classe.
Nome DEVE bater com o do arquivo:
20170910234643_create_contacts.rb

```
class CreateContacts < ActiveRecord::Migration[5.1]
  def change
    create_table :contacts do |t|
      t.string :name
      t.string :email
      t.text :remark
      t.references :kind, foreign_key: true

      t.timestamps
    end
  end
end
```

"Criar uma tabela chamada contacts"

Atributo name do tipo String
Perceba que é um método:
t.string(:name)

Método especial.
Cria campos de "criado em" e "atualizado em" na tabela

INSTITUTO FEDERAL

Tabela Gerada

- Verificar no sqlite3 por linha de comando

```
diego@diego-avell ~/projetos/rails/agenda $ sqlite3 db/development.sqlite3
SQLite version 3.11.0 2016-02-15 17:29:24
Enter ".help" for usage hints.
sqlite> PRAGMA table_info(contacts);
0|id|INTEGER|1||1
1|name|varchar|0||0
2|email|varchar|0||0
3|remark|text|0||0
4|kind_id|integer|0||0
5|created_at|datetime|1||0
6|updated_at|datetime|1||0
sqlite>
```

INSTITUTO FEDERAL

Migrações

- Com migrações, logo “perdemos o controle” sobre como está nosso modelo de dados
  - Migrações são uma sequência de scripts que devem ser executados em ordem
- Por isso existe o arquivo:
  - db/schema.rb
    - Criado pelo comando db:schema:dump
      - Chamado automaticamente após o rails db:migrate

INSTITUTO FEDERAL

Migrações - Regras

- **Toda alteração de BD no Rails deve ser feita através de uma Migration**
  - Criar tabelas, adicionar campo, mudar tipo de dados de campo, deletar tabela ou campo, …
- **Alteração livre no arquivo da migração**
  - Somente antes de executar (rails db:migrate)
- **Nunca alterar uma Migração "antiga"**
  - Ela pode já ter sido executada no BD e não será revisitada

INSTITUTO FEDERAL

Model

- Todo model no Rails deve ficar na pasta:
  - app/models
- Nome do arquivo:
  - Nome da classe em snake_case
    - contact.rb
    - contact_info.rb
- É uma instância de ApplicationRecord
  - Que é uma instância de ActiveRecord::Base

INSTITUTO FEDERAL

ActiveRecord

- ActiveRecord é o framework de ORM (Object-relational mapping) usado pelo Rails
- Ele é quem faz a “mágica” da conversão de dados no BD para Objetos Ruby e vice-versa
- Fornece modelos, consultas, métodos de persistência, controle de transação, …
  - Sempre trabalharemos exclusivamente OO

INSTITUTO FEDERAL

Cara de um Model

- Por enquanto, temos models bem simples
  - Mas já funcionais

```
class Contact < ApplicationRecord
  belongs_to :kind
end


class Phone < ApplicationRecord
  belongs_to :contact
end
```

Métodos auxiliares para mapear como Objeto Ruby,
pois no BD temos somente kind_id e contact_id (inteiros).
Desta forma poderemos fazer:
contact.kind ← Objeto Ruby
phone.contact ← Objeto Ruby

INSTITUTO FEDERAL

Controller

- Todo controller no Rails deve ficar na pasta:
  - app/controllers
- Nome do arquivo:
  - [snake_case no plural]_controller.rb
    - contacts_controller.rb
    - kinds_controller.rb
- É uma instância de ApplicationController
  - Que é uma instância de ActionController::Base

INSTITUTO FEDERAL

Ações

- Métodos públicos dentro de controllers PODEM representar ações
  - Diretamente ligadas às rotas (URL)
- Fluxo padrão do MVC no Rails:
  - Rota recebe a requisição
  - Dispara para um controller/action
  - A action é um método no controller que toma as decisões e faz o que precisa ser feito

INSTITUTO FEDERAL

Ações Padrão

- O scaffold já cria um conjunto de ações padrão:
  - **index**: ação padrão → /contacts [GET]
  - **show**: mostrar um → /contacts/1 [GET]
  - **new**: form (novo) → /contacts/new [GET]
  - **edit**: form (edição) → /contacts/edit [GET]
  - **create**: trata form (novo) → /contacts [POST]
  - **update**: trata form (edição) → /contacts/1 [PUT]
  - **destroy**: remove um item → /contacts/1 [DELETE]
- Notou algo?
  - CRUD no padrão REST?

INSTITUTO FEDERAL

Rotas

- As Rotas são as portas de entrada da nossa aplicação Rails
- Sem rotas, um controllers não têm sentido
- Mapeamento:
  - URL → Controller (action)
- Arquivo de rotas:
  - config/routes.rb

INSTITUTO FEDERAL

Rotas Padrão

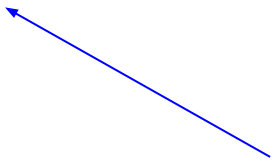
- Quando usamos o Scaffold, ganhamos um conjunto de rotas padrão
  - Aquelas descritas nas ações
    - /contacts/1 …
- Esta é só uma forma de trabalhar
  - Podemos criar as URLs (internas) que quisermos para nossa aplicação
    - /minha_tela/ola_mundo
    - /chuchu/teste123.json

INSTITUTO FEDERAL

Rotas Padrão

- Arquivo routes.rb criado com o scaffolding:

```
Rails.application.routes.draw do
  resources :phones
  resources :addresses
  resources :contacts
  resources :kinds
end
```

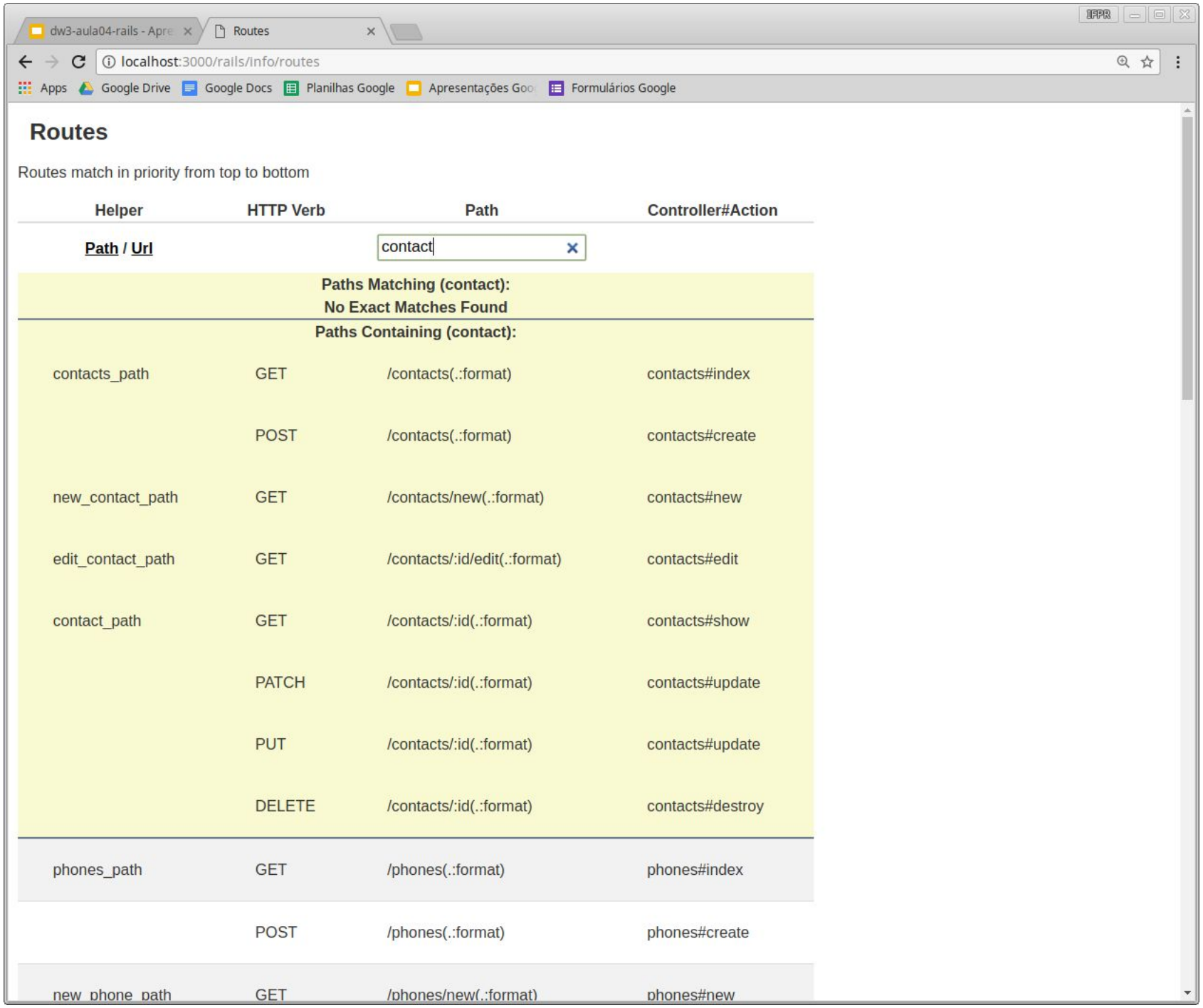


O método resources provê um mapeamento implícito de rotas para as ações:
index, show, new, edit, create, update, delete

INSTITUTO FEDERAL

Verificar Rotas

- Podemos verificar (em tempo real) quais são todas as rotas válidas para nossa aplicação
- Acessar URL:
  - http://server/rails/info/routes
- Ou no terminal
  - rails routes



Routes

Routes match in priority from top to bottom

Helper	HTTP Verb	Path	Controller#Action
Path / Url		contact	
		Paths Matching (contact): No Exact Matches Found	
		Paths Containing (contact):	
contacts_path	GET	/contacts(.:format)	contacts#index
	POST	/contacts(.:format)	contacts#create
new_contact_path	GET	/contacts/new(.:format)	contacts#new
edit_contact_path	GET	/contacts/:id/edit(.:format)	contacts#edit
contact_path	GET	/contacts/:id(.:format)	contacts#show
	PATCH	/contacts/:id(.:format)	contacts#update
	PUT	/contacts/:id(.:format)	contacts#update
	DELETE	/contacts/:id(.:format)	contacts#destroy
phones_path	GET	/phones(.:format)	phones#index
	POST	/phones(.:format)	phones#create
new_phone_path	GET	/phones/new(.:format)	phones#new

INSTITUTO FEDERAL

Brincando com as Rotas

- Podemos usar os parâmetros :only ou :except para definir rotas liberadas/bloqueadas
  - `resources :kinds, except: :destroy`
    - Todas rotas (ações) de kinds, exceto a destroy
  - `resources :phones, only: [:index, :new, :create]`
    - Apenas as ações index, new e create de phones

- Testar no navegador

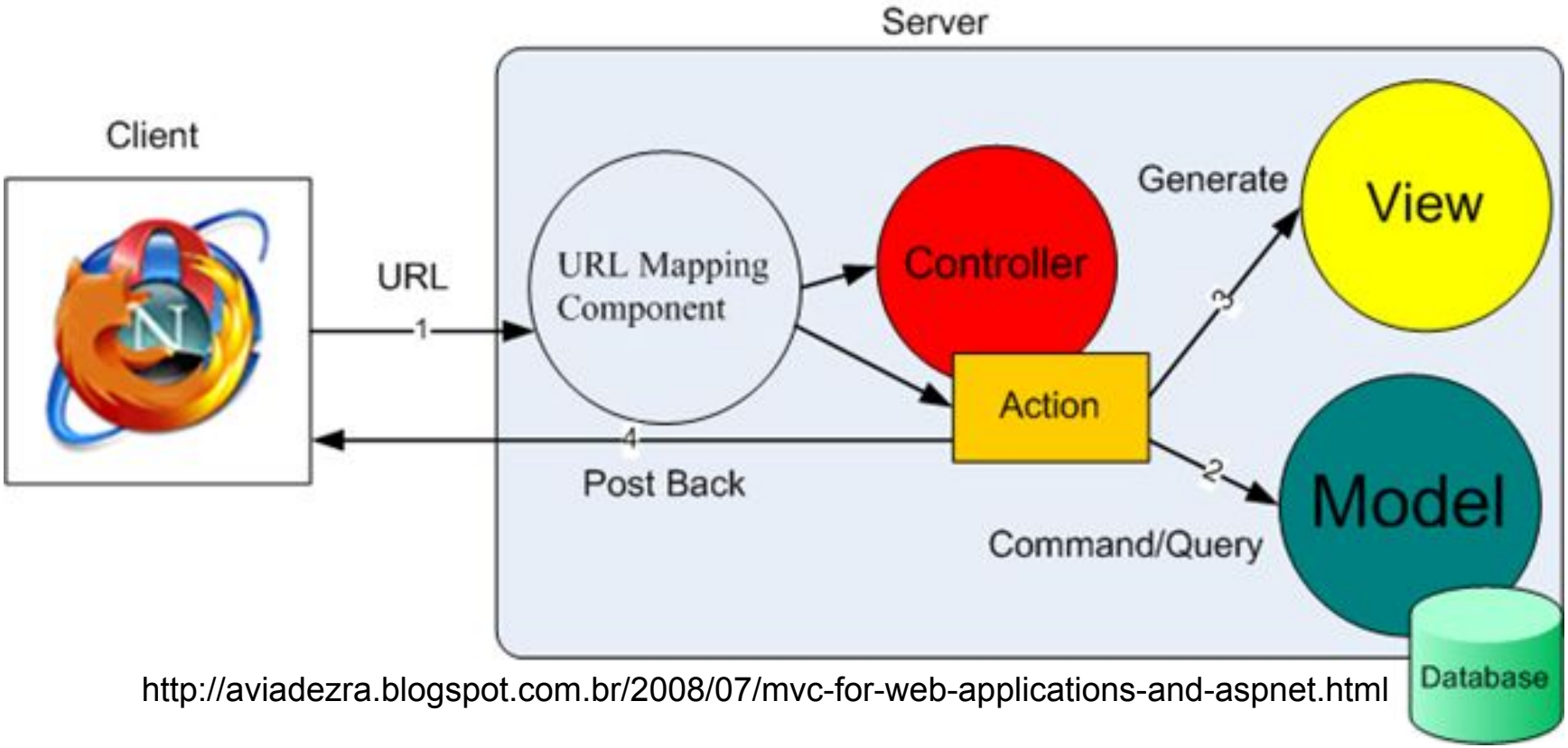
INSTITUTO FEDERAL

Views

- Uma view pode ser qualquer forma de visualizar (saída) um dado provido por determinado controller/action
- Forma padrão:
  - ERB (Embedded Ruby)
    - Marcação HTML com trechos de código Ruby
- Controllers e Views possuem uma ligação íntima e automática
  - Convenção sobre Configuração

Fluxo de Requisições (MVC)

- Uma Action gera (renderiza) determinada View



http://aviadezra.blogspot.com.br/2008/07/mvc-for-web-applications-and-aspnet.html

INSTITUTO FEDERAL

Views

- Diretório das views
  - app/views
- Dentro temos diretórios para cada model
  - Dentro destes, temos suas views
    - Arquivos .html.erb (Embedded Ruby)
- Exemplos:
  - app/views/contacts/index.html.erb
  - app/views/kinds/show.html.erb
  - app/views/addresses/new.html.erb

INSTITUTO FEDERAL

Renderização

- Em cada ação dos controllers
  - Precisa-se dizer o que deve ser renderizado e como renderizar
- Se não for informado, procura por view padrão
- Exemplo:
  - Método show, dentro de KindsController

```
def show
end
```

  - Vai automaticamente renderizar a view:
    - app/views/kinds/show.html.erb

INSTITUTO FEDERAL

Marcação ERB

- Dentro de um arquivo .html.erb teremos duas formas de inserção de código Ruby
- `<% codigo_ruby %>`
  - Executa um código Ruby qualquer
    - Pode ser uma abertura ou fechamento de bloco
- `<%= codigo_ruby_para_imprimir %>`
  - A sintáxe com o caractere de igual (=), vai imprimir a saída no HTML

INSTITUTO FEDERAL

Atributos na View

- Uma view tem acesso a todos os atributos internos do controller
  - Como se fosse parte do objeto do controller
- Exemplo:

KindsController

```
def index
  @kinds = Kind.all
end
```

views/kinds/index.html.erb

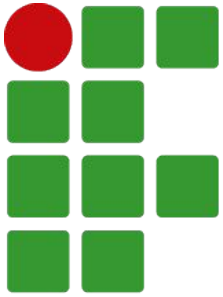
```
<% @kinds.each do |kind| %>
   <%= kind.description %>
<% end %>
```

INSTITUTO FEDERAL

Marcação ERB - Exemplo

- contacts/index.html.erb

```
<tbody>
  <% @contacts.each do |contact| %>
    <tr>
      <td><%= contact.name %></td>
      <td><%= contact.email %></td>
      <td><%= contact.remark %></td>
      <td><%= contact.kind %></td>
      <td><%= link_to 'Show', contact %></td>
      <td><%= link_to 'Edit', edit_contact_path(contact) %></td>
      <td><%= link_to 'Destroy', contact, method: :delete,
                     data: { confirm: 'Are you sure?' } %></td>
    </tr>
  <% end %>
</tbody>
```



INSTITUTO FEDERAL
Paraná
Campus Paranaguá

Projeto Agenda de Contatos

Melhorando Alguns Recursos

INSTITUTO FEDERAL

Problemas

- Navegando pela nossa aplicação gerada pelos scaffoldings, notaremos algumas “falhas”
  - Não existe página principal (digitar URL)
  - Precisamos informar o código do elemento nos formulários com associações
  - Falta de um template de navegação

INSTITUTO FEDERAL

Criando Controller

- Vamos criar um novo controller para alojar a nossa nova página principal
  - Será a nossa Home
- Usar comando:
  - rails generate controller <nome> <action>
- Digitar
  - rails g controller home index

INSTITUTO FEDERAL

HomeController

- O comando gera para nós:
  - app/controllers/home_controller.rb
    - Contém classe HomeController
  - Classe HomeController com método
    - index
      - (Vazio, mas já renderizando view padrão)
  - app/views/home/index.html.erb
    - View Padrão
  - Rota get 'home/index'
    - http://server/home/index
  - Helpers, assets, testes, …

INSTITUTO FEDERAL

Definir Página Inicial

- Precisamos dizer qual controller/action chamar quando nenhum é informado
  - No caso, abrir como página inicial
- No arquivo routes.rb
  - Após `get 'home/index'`, adicionar:
    - `root 'home#index'`

INSTITUTO FEDERAL

Prática

- Edite a view da página inicial
  - Coloque uma mensagem de boas vindas
  - Adicione links para todos os cadastros (index)

INSTITUTO FEDERAL

Lista de Seleção

- No navegador, abra o formulário de cadastro de contatos (/contacts/new)
- Note que Kind precisa ser digitado
  - Interessante seria o usuário escolher (lista)
- Alterar app/views/contacts/_form.html.erb
  - Trocar:
    - `<%= form.text_field :kind_id, id: :contact_kind_id %>`
  - Por:
    - `<%= collection_select :contact, :kind_id, Kind.all, :id, :description %>`

INSTITUTO FEDERAL

Prática

- Colocar o collection select nas demais views de formulário

INSTITUTO FEDERAL

collection_select

- `collection_select(object, method, collection, value_method, text_method)`
  - Monta um select - option
    - object: tipo base (onde gravar)
    - method: atributo a receber value selecionado
    - collection: lista de objetos para escolha
    - value_method: atributo id para cada item da lista
    - text_method: atributo a ser mostrado para cada item da lista

INSTITUTO FEDERAL

Helpers

- Helper são métodos que auxiliam as views no processo de renderização
  - Podemos criá-los, mas o Rails dispõe muitos
- collection_select ← Renderiza um <select>
- form_with ← Renderiza um <form>
- link_to ← Renderiza um <a> - Muito usado
- render ← Renderiza um partial ERB

INSTITUTO FEDERAL

Melhorando Elemento

- Uma forma um pouco mais bonita de fazer usar o `collection_select`
  - Usando a API do `form_with`
- Adapte este trecho no código e entenda
  - Por que é mais bonito?

```
<div class="field">
  <%= form.label :kind %>
  <%= form.collection_select :kind_id, Kind.all, :id, :description, id: :contact_kind %>
</div>
```

INSTITUTO FEDERAL

Template

- Percebeu que todas as views contém somente um pedaço de HTML
- Todas views são trechos de código renderizados dentro de um template
- O template está no arquivo:
  - views/layouts/application.html.erb
- Abra o arquivo e procure por **`<%= yield %>`**
  - Este é o trecho que será substituído pela view que se deseja mostrar

INSTITUTO FEDERAL

Prática

- Arrume o template para apresentar algum conteúdo padrão relevante para todas as páginas, como:
  - Cabeçalho
  - Rodapé
  - Link para página inicial e outros
  - …

INSTITUTO FEDERAL

Prática

- As views de index e show de contacts, phones e addresses mostram algo estranho:
  - #<Kind:0x007f649c0da018>
- O que é isso?
- Como poderíamos arrumar?