



Ruby on Rails

Suzyanne Oliveira

Construindo aplicação
web com scaffold.

- Preciso de um grande favor! Recorda-se daquela aplicação de venda de ingressos na qual eu disse que estava trabalhando ? Não está funcionando direito. Já passamos uma semana trabalhando nisso! A equipe está tendo sérios problemas.

“Você acha que consegue criar uma nova aplicação para nós ? ”



Nós precisamos de um site que possa :

- Listar todos os ingressos vendidos;
 - Criar uma nova venda de ingressos;
 - Ler e exibir um ingresso único;
 - Atualizar as informações de venda;
 - Excluir uma venda de ingresso.
-

The front page will need to list all of the tickets that have been sold.

There will be a button on the front page that will let you create a new ticket sale.

Hand-drawn sketch of a web browser window titled "New ticket". The browser address bar shows "http://localhost:3000/tickets/new". The form contains the following fields and controls:

- Name:
- Ticket #:
- Address:
- Price paid:
- Email address:
- Buttons: "Create" and "Back"

As well as a "Show" link, there will be an "Edit" link that can be used to update the details of a ticket sale.

Hand-drawn sketch of a web browser window titled "Editing ticket". The browser address bar shows "http://localhost:3000/tickets/1/edit". The form contains the following fields and controls:

- Name:
- Ticket #:
- Address:
- Price paid:
- Email address:
- Buttons: "Update" and "Show / Back"

Hand-drawn sketch of a web browser window titled "Listing tickets". The browser address bar shows "http://localhost:3000/tickets". The table lists the following data:

Name	Ticket #	Address	Price paid	Email address	
Alan Longman	1A	17 Hensbury Road, Ashford, MS	40.0	alan@longman.com	Show Edit Delete
Gordon Clark	956	17 Tudor Street, Cambridge, MA	95.0	gordon@clark.com	Show Edit Delete
Bobby Lloyd	98C	9 Main Street, Portsmouth, RI	45.0	bobby@lloyd.com	Show Edit Delete
Eric Henshaw	20	1127 Main Ave, Cambridge, MA	27.0	eric@henshaw.org	Show Edit Delete
Kelly Henderson	88U	445 Townsend St, Boston, MA	49.0	kelly@henderson.org	Show Edit Delete

Below the table is a "New ticket" link.

Next to every ticket on the list there will be a "Show" link that will display the details of a single ticket.

Hand-drawn sketch of a web browser window showing the details of a single ticket. The browser address bar shows "http://localhost:3000/tickets/1A". The details are:

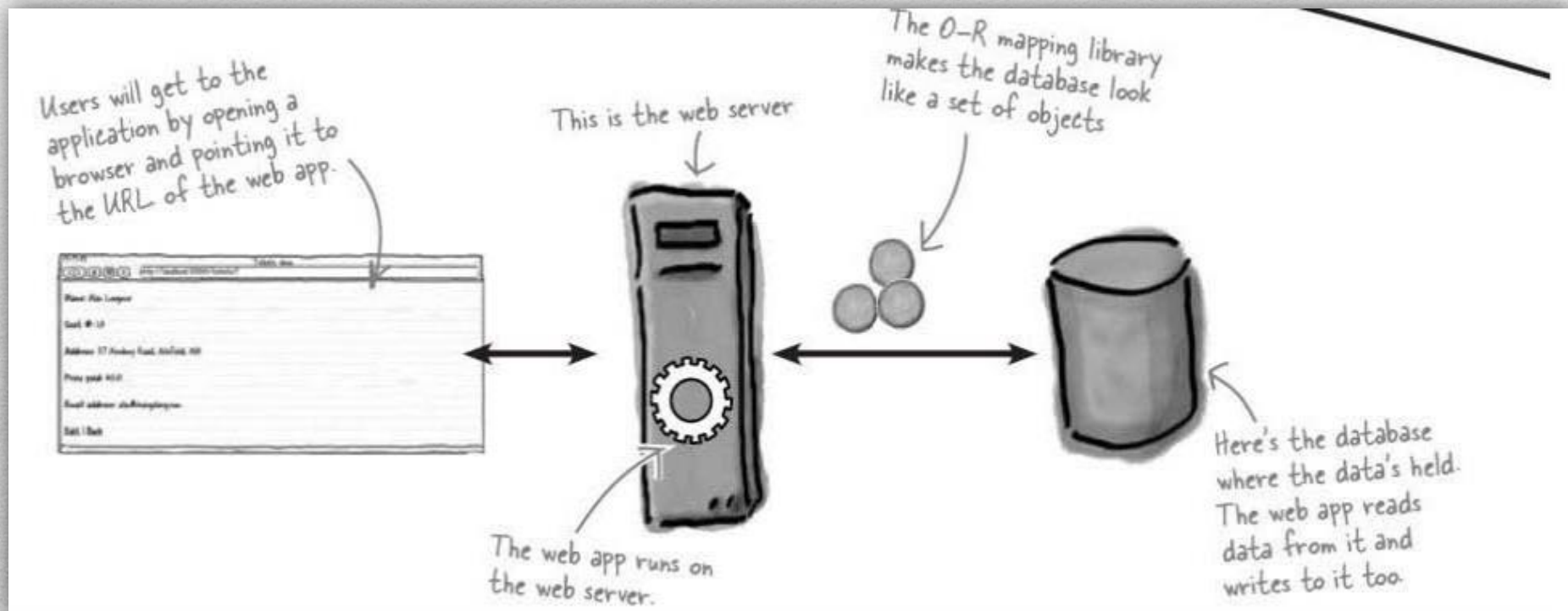
- Name: Alan Longman
- Ticket #: 1A
- Address: 17 Hensbury Road, Ashford, MS
- Price paid: 40.0
- Email address: alan@longman.com
- Buttons: "Show" and "Back"



Finally there will be a "Delete" link to remove a ticket sale.

Então o que precisamos para a aplicação ?

- Um framework de aplicação
- Um sistema de banco de dados
- Um servidor web
- Uma biblioteca de mapeamento objeto-relacional



Então ,como o RAILS nos ajuda?

- Independente da linguagem que você usa para desenvolver seu código, provavelmente precisará desses três recursos para ter sua aplicação totalmente implantada. Uma das melhores qualidades do rails é que ele contém tudo o que você precisa para seu software (tudo agrupado gratuitamente).
-

Coisas que você vai precisar saber sobre o rails :

- A aplicação web precisará permitir ao usuário acessar e modificar dados ,para isso o Rails inclui um framework de aplicação chamado ACTION PACK ,que lhe ajudará a gerar dados e páginas interativas.
 - O Rails tem um banco chamado SQLite3.
 - E o Rails tem um servidor inteiro chamado WebBrink.
 - Outra coisa é que você irá precisar de uma biblioteca de mapeamento objeto-relacional e o Rails fornece uma chamada ACTIVE RECORD.
-



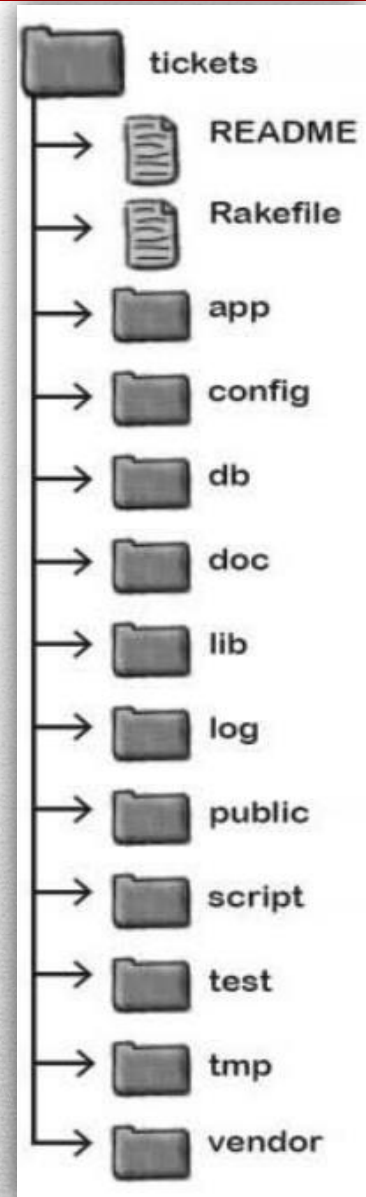
Criando aplicação tickets

Abrir o terminal e digitar :

- Tickets é o nome do seu projeto no Rails

rails new tickets

- O Rails gera todo um conjunto de outros diretórios e arquivos que formam uma estrutura básica de uma aplicação.
- Isso significa que você efetivamente criou uma aplicação web básica inteira apenas com um comando.



- Acabamos de criar uma aplicação web ,precisamos reiniciar o servidor web inteiro para ver se a aplicação está executando.
- No terminal, mude para dentro da pasta Tikects e digite :

Rails server

```
PS C:\Users\suzyanne> cd ~
PS C:\Users\suzyanne> cd Documents
PS C:\Users\suzyanne\Documents> ls
```

Diretório: C:\Users\suzyanne\Documents

Mode	LastWriteTime	Length	Name
d-----	15/03/2013 12:32		Add-in Express
d-----	10/04/2013 15:34		aplicacao2
d-----	17/03/2013 14:21		CarbonCS v1.1
d-----	13/03/2013 19:51		Downloads
d-----	19/03/2013 13:36		sistemas
-a-----	18/03/2013 20:08	73293	1.PNG
-a-----	18/03/2013 20:09	12008	2.PNG
-a-----	18/03/2013 20:09	20929	3.PNG
-a-----	18/03/2013 20:10	18591	4.PNG
-a-----	18/03/2013 16:37	1796	bkn.jpg
-a-----	10/04/2013 19:22	33071	questao.PNG
-a-----	07/04/2013 19:16	1510	soma_mult.alg
-a-----	18/03/2013 16:44	246	testelayout.html
-a-----	10/04/2013 20:08	222306	tpp.PNG

```
PS C:\Users\suzyanne\Documents> cd sistemas
PS C:\Users\suzyanne\Documents\sistemas> ls
```

Diretório: C:\Users\suzyanne\Documents\sistemas

Mode	LastWriteTime	Length	Name
d-----	18/03/2013 15:09		app
d-----	18/03/2013 15:09		config
d-----	18/03/2013 21:13		db
d-----	18/03/2013 15:09		doc
d-----	18/03/2013 15:09		lib
d-----	18/03/2013 15:12		log
d-----	19/03/2013 13:36		public
d-----	18/03/2013 15:09		script
d-----	18/03/2013 15:09		test
d-----	19/03/2013 13:23		tmp
d-----	18/03/2013 15:09		vendor
-a-----	18/03/2013 15:09	430	.gitignore
-a-----	18/03/2013 15:09	158	config.ru
-a-----	18/03/2013 15:09	766	Gemfile
-a-----	18/03/2013 15:09	2548	Gemfile.lock
-a-----	18/03/2013 15:09	273	Rakefile
-a-----	18/03/2013 15:09	9208	README.rdoc

```
PS C:\Users\suzyanne\Documents\sistemas> rails server
```

- Abra seu navegador e digite : **localhost:3000**
- O padrão é a porta 3000 se quiser executar em outra porta
rails server -p 8000



1. APLICAÇÃO RAILS SEMPRE SEGUEM CONVERÇÕES

- Todas as aplicações seguem a mesma estrutura básica de arquivos e utilizam nomes e consistentes para tudo. Isso torna a aplicação mais fácil de entender, mas também significa que as ferramentas internas do rails entenderão como as aplicações funcionam.
-

Perguntas ?

- **P: Falando de Ruby e Rails. Qual a diferença entre os dois?**

R: Ruby é uma linguagem de programação. Rails é uma coleção de Scripts Ruby. Assim, o servidor web, o framework de aplicação ActionPack e os scripts de ferramenta embutidos são todos scripts Ruby, portanto ,parte do rails.

- **P: Como edito a página principal do meu site?**

R: Usando o arquivo HTML no arquivo public/index.html, abaixo do diretório da aplicação. O diretório público contém todo o conteúdo estático para a aplicação.

- **P: Faz diferença a pasta em que estou ao executar rails server ?**

R: Sim, claro. Você precisa estar na pasta que contém a sua aplicação web.

- **P: Isto é o que compila o meu código ?**

R: Ruby é uma linguagem interpretada ,como JavaScript . Isto significa que não é necessário compilação.

- **Nós precisamos de um site que possa :**
 - **Listar todos os ingressos vendidos;**
 - **Criar uma nova venda de ingressos;**
 - **Ler e exibir um ingresso único;**
 - **Atualizar as informações de venda;**
 - **Excluir uma venda de ingresso.**

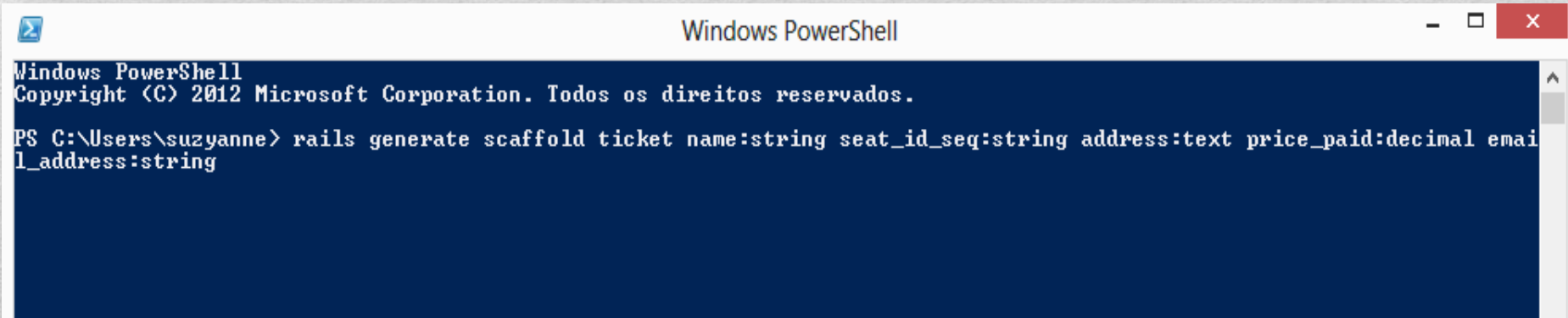


- **Nossa estrutura de Dados**
 - **name** : Nome do Computador(String)
 - **seat-id-seq** : O numero do lugar(String)
 - **address** : Endereço do comprador(string)
 - **price_paid**: Valor do ingresso(decimal)
 - **email_adress**:Email do Comprador(string)
-

- Então precisamos criar páginas web que nos permitam **CRIAR,LER,ATUALIZAR** e **EXCLUIR** ingressos. As letras iniciais em inglês de cada operação são **CRUD**(*create, read, update, delete*), por isso são conhecidos como operações de **CRUD**.
 - São operações muito comuns nas aplicações de banco de dados – Tão comuns que o Rails tem um jeito de gerar, de forma mais rápida, todo código e todas as páginas de que você precisa. Ele faz tudo isso utilizando um comando **SCAFFOLD**.
-


Maís o que realmente o comando scaffold faz ?

- O scaffold cria o código que permitirá criar, ler, atualizar e excluir dados, o scaffold consegue economizar muito do seu tempo e esforço.

A screenshot of a Windows PowerShell terminal window. The title bar at the top says "Windows PowerShell" and has standard window controls (minimize, maximize, close). The terminal content shows the Windows PowerShell logo, a copyright notice for 2012 Microsoft Corporation, and a command being entered: "rails generate scaffold ticket name:string seat_id_seq:string address:text price_paid:decimal email_address:string".

```
Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. Todos os direitos reservados.

PS C:\Users\suzyanne> rails generate scaffold ticket name:string seat_id_seq:string address:text price_paid:decimal email_address:string
```

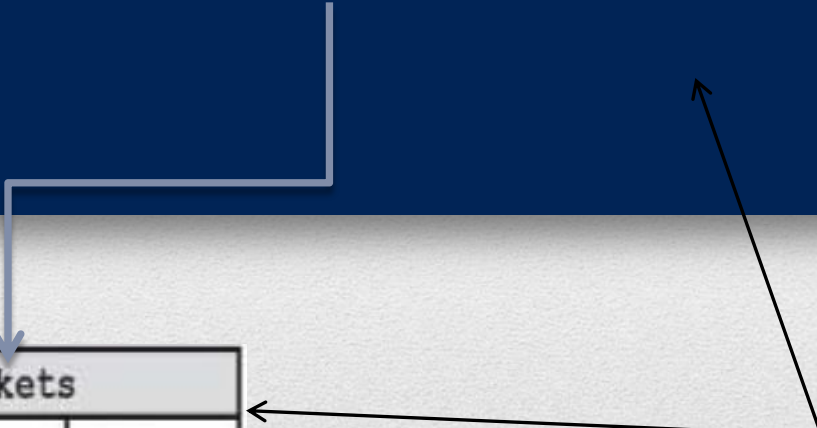
**Por que você acha que a
execução da aplicação deu
errado?**

Ainda não há tabelas no banco de dados !

- A aplicação deveria ter exibido uma lista vazia de ingressos vendidos, mas não faz isso. POR QUE NÃO?
 - Ela precisa ler a lista da tabela chamada **tckites** , no banco de dados, porém ainda não criamos nenhuma tabela no banco.
-


```
Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. Todos os direitos reservados.

PS C:\Users\suzyanne> rails generate scaffold ticket name:string seat_id_seq:string address:text price_paid:decimal email_address:string
```

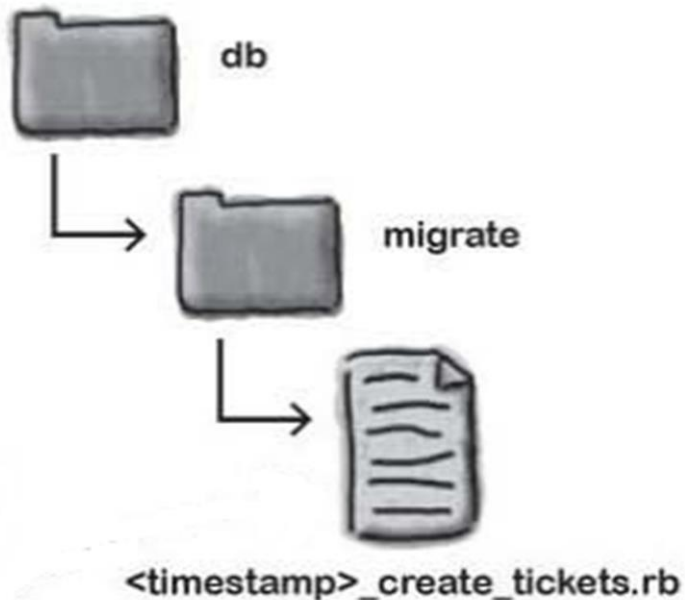


tickets	
name	string
seat_id_seq	string
address	text
price_paid	decimal
email_address	string

NOTA: O comando scaffold refere-se a ticket e a tabela será chamada de tickets(no plural)

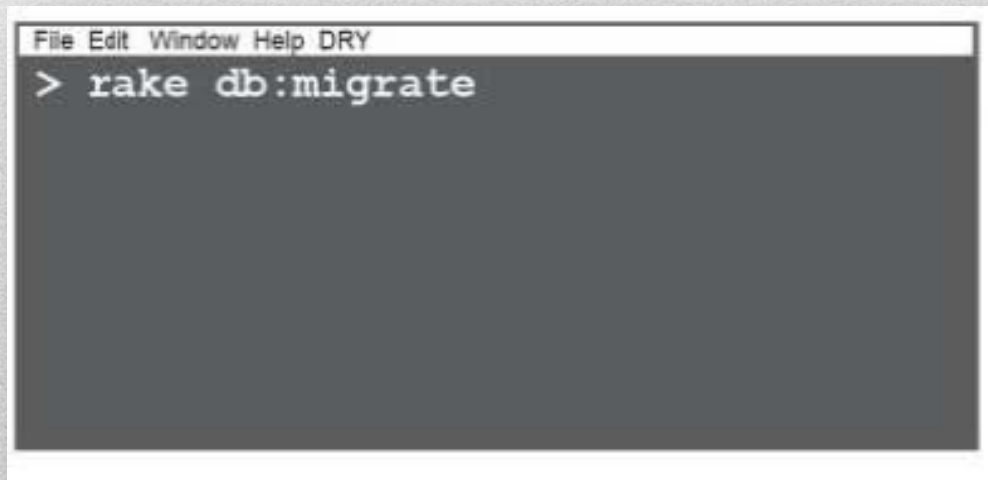
Então , como podemos fazer o rails criar uma tabela?

- Quando o rails gerar uma aplicação usando o scaffold ,também será gerado um pequeno script ruby chamado **migration** para criar a tabela.
- Uma migração (mudança) é um script que altera a estrutura subjacente do banco de dados.



```
class CreateTickets < ActiveRecord::Migration
  def self.up
    create_table :tickets do |t|
      t.string :name
      t.string :seat_id_seq
      t.text :address
      t.decimal :price_paid
      t.string :email_address
      t.timestamps
    end
  end
  def self.down
    drop_table :tickets
  end
end
```


- O **migration** é um script Ruby pequeno. Ao invés de executar este script de forma direta, você deve executar utilizando outra ferramenta do rails, chamada de **RAKE**.
- Para executar a migração ,digite **rake db:migrate** no terminal .Isto executa o código de migração e criará a tabela.

A screenshot of a terminal window with a dark gray background and a white border. The title bar at the top contains the text 'File Edit Window Help DRY'. The main area of the terminal shows a prompt character '>' followed by the command 'rake db:migrate' in a white monospaced font.

```
File Edit Window Help DRY
> rake db:migrate
```

- Antes de tudo vá na sua aplicação e remova o arquivo :
→ *view/layouts/application.html.erb*
- Agora abra seu navegador e atualize a página
- *https://localhost:3000/tickets*



Sistema de Ingressos Online

Ingressos Cadastro:

Name	Lugar	Endereco	Valor Pago	Email	Phone			
suzyanne	76r	rua da concordia	80.0	suzy_0310@yahoo.com.br	45645645	Visualizar	Editar	Deletar
Mateus	876R	rua peneiras	9000.0	matheus_oliveira@hotmail.com	982748972985	Visualizar	Editar	Deletar
Rebeca Oliveira	576	rua vila lobos	900.0	rebeca@hotmail.com	89758937598	Visualizar	Editar	Deletar
mateus de oliveira queiroz	4757hfdhf	rua latino	7990.0	matheus@yahoo.com.br	988923748092743	Visualizar	Editar	Deletar
suile	34534hyhfsd	rua avinada tal	342.0	suile@htotmail.com	892748947832432	Visualizar	Editar	Deletar
Lucas	6753r	rua brasil	897.0	lucas@hotmail.com	29873487423	Visualizar	Editar	Deletar
ana	iqew8979wq	rua maria Lacerda	287.0	ana@hotmail.com	984892734	Visualizar	Editar	Deletar

[Novo Cadastro](#)



Só inserimos alguns comandos no console e isso criou todas as aplicações ?

- Sim ,criamos muito mais do que uma simples página inicial. Criamos todo um sistema.
- O scaffold gerou todo um conjunto de páginas que nos permite criar ,modificar, atualizar e excluir informações dos ingressos.

Nome

Lugar

Endereco

Valor

Email

Phone

Create Sistema

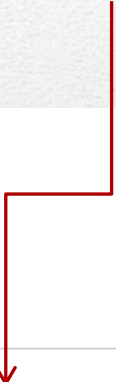
[Voltar](#)

❑ TELA DE CADASTRO

- Clicando no (**create sistema**) vai ser direcionado a um formulário onde podemos criar um novo ingresso.

❑ TELA DE VISUALIZAÇÃO DO CADASTRO

- Quando você envia o formulário , conseguirá ler o novo ingresso ,que retornou do banco de dados, e exibi-los .



Nome: suzyanne

Lugar: 76r

Endereco: rua da concordia

Valor Pago: 80.0

Email: suzy_0310@yahoo.com.br

Phone: 45645645

[Editar](#) | [Voltar](#)

A screenshot of a web form titled "TELA DE EDITAR O CADASTRO". The form contains several input fields for user information: "Nome" (suzyanne), "Lugar" (76r), "Endereco" (rua da concordia), "Valor" (80), "Email" (suzy_0310@yahoo.com.br), and "Phone" (45645645). Below the input fields is a button labeled "Update Sistema". At the bottom of the form are two links: "Visualizar" and "Voltar".

Nome
suzyanne

Lugar
76r

Endereco
rua da concordia

Valor
80

Email
suzy_0310@yahoo.com.br

Phone
45645645

Update Sistema

[Visualizar](#) | [Voltar](#)

❑ TELA DE EDITAR O CADASTRO

- O botão editar ,na página de exibição, permite – lhe atualizar qualquer informação que você quiser.

Revisão ...

❑ O comando **rails <nome da aplicacao>** gera uma aplicação web na pasta **<appname>**. O rails cria também as pastas e arquivos que formam a estrutura básica da sua aplicação.

❑ O rails vem com um servidor http embutido. Para inicializar o servidor, usamos o comando **RAILS SERVER**.

❑ A home page padrão pode ser acessada assim:

<http://localhost:3000/>

❑ A aplicação do rails seguem a ideia de conversão sobre configuração.

Criar ,ler,atualizar e excluir no banco de dados são conhecidas como operações CRUD.

❑ SCAFFOLD gera o código CRUD para você .Comando é rails generate scaffold “X” <nome da coluna>:<tipo da coluna>

❑ Para visualizar o seu scaffold, digite a URL abaixo do navegador : **http://localhost:3000/<tabela no plural>**

❑ Uma migração é um script que altera a estrutura do banco de dados selecionado.Você executa a migração com o comando : **rake db:migrate**

Perguntas ?

- **P:Alguns comandos começam com rails, outros começam com ruby e alguns com rake. Qual a diferença?**

R: O comando Rails é utilizado para criar uma nova aplicação. O ruby é o interpretador Ruby e é utilizado para executar scripts das ferramentas, que são armazenadas na pasta scripts. Os comandos ruby e rake são utilizados praticamente para tudo no rails.

- **P:O que é o rake?**

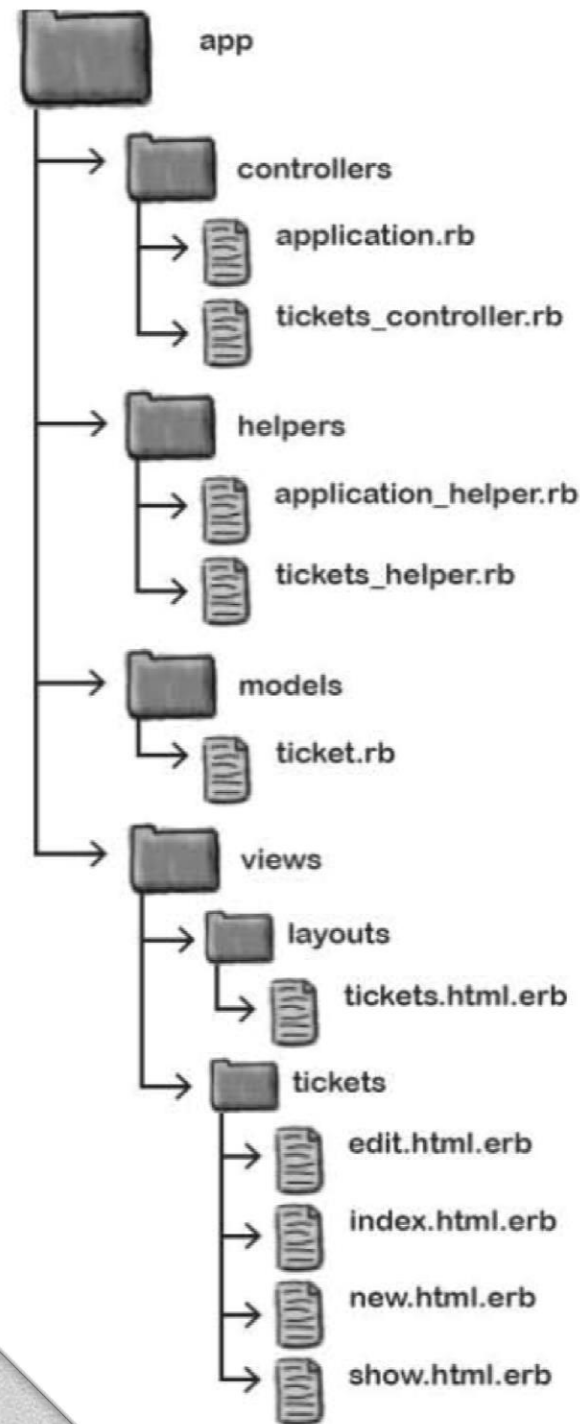
R:É o comando utilizado para executar a migração do banco de dados. O nome significa “ruby faz”.

- **P:Convenção sobre configuração ,como assim?**

R: O rails tem uma abordagem diferente. Esses elementos no rails são nomeados de forma consistente e armazenados em lugares padronizados. Isto se chama abordagem “convencional” .

Para modificar uma aplicação, precisamos entender a arquitetura.

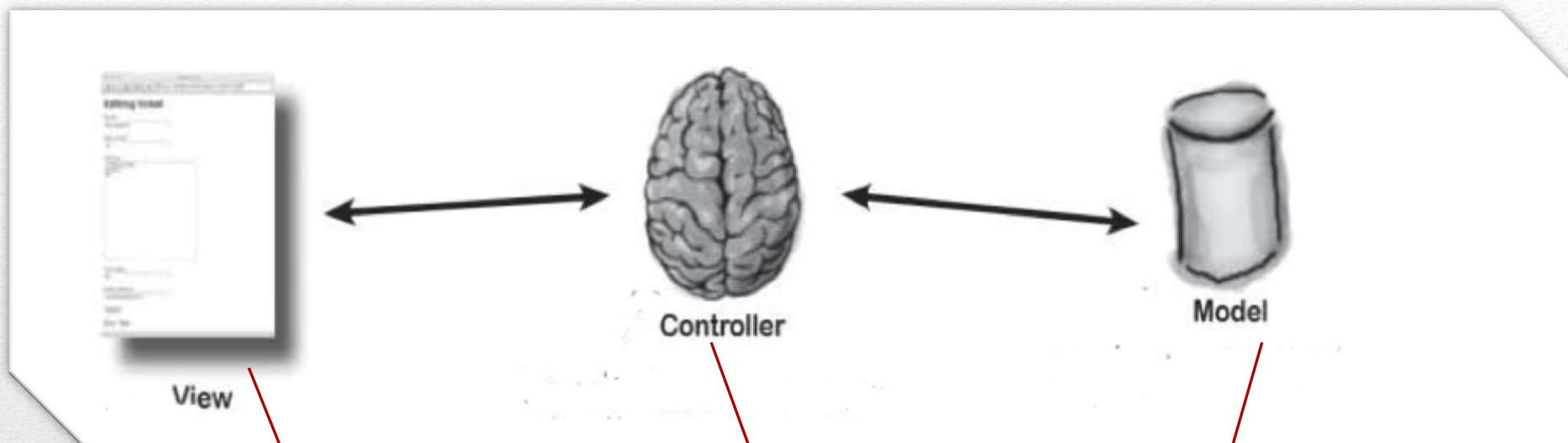
- O comando scaffold apenas gera o código básico para nós. Uma vez gerado o responsável pela personalização é você.
- Veja a pasta APP, nela há muitos códigos gerados para personalizar.



MVC – MODELO-VISUALIZAÇÃO-CONTROLE

1. **Código Modelo(Model):** O código modelo gerencia como os dados escritos e lidos no seu banco de dados. Os objetos do código modelo representam coisas que existem no domínio do problema do sistema – como ingressos no sistema de tickets.
2. **Código de visualização(View):** A visualização ou “visão” é a parte da aplicação que é representada pelo usuário. Por isso, às vezes é chamada de camada de apresentação.
3. **Código Controlador(Controller) :** O controlador é o verdadeiro cérebro da aplicação. Ele decide como o usuário interage com o sistema, controlando quais dados são acessados a partir do modelo(model) e quais partes da visualização(view) vão apresentá-los.





A visualização que consiste nas páginas que apresentam a aplicação ao usuário.

O controlador é a chave que toma decisão. Ele decide quais dados precisam ser acessados a partir do modelo(model) e quais partes da visualização(view) vão apresentá-los .

O modelo simula as coisas no mundo real que a aplicação está gerenciando. No sistema de vendas de ingressos ,os objetos "ingressos"(tickets) ficam no modelo.

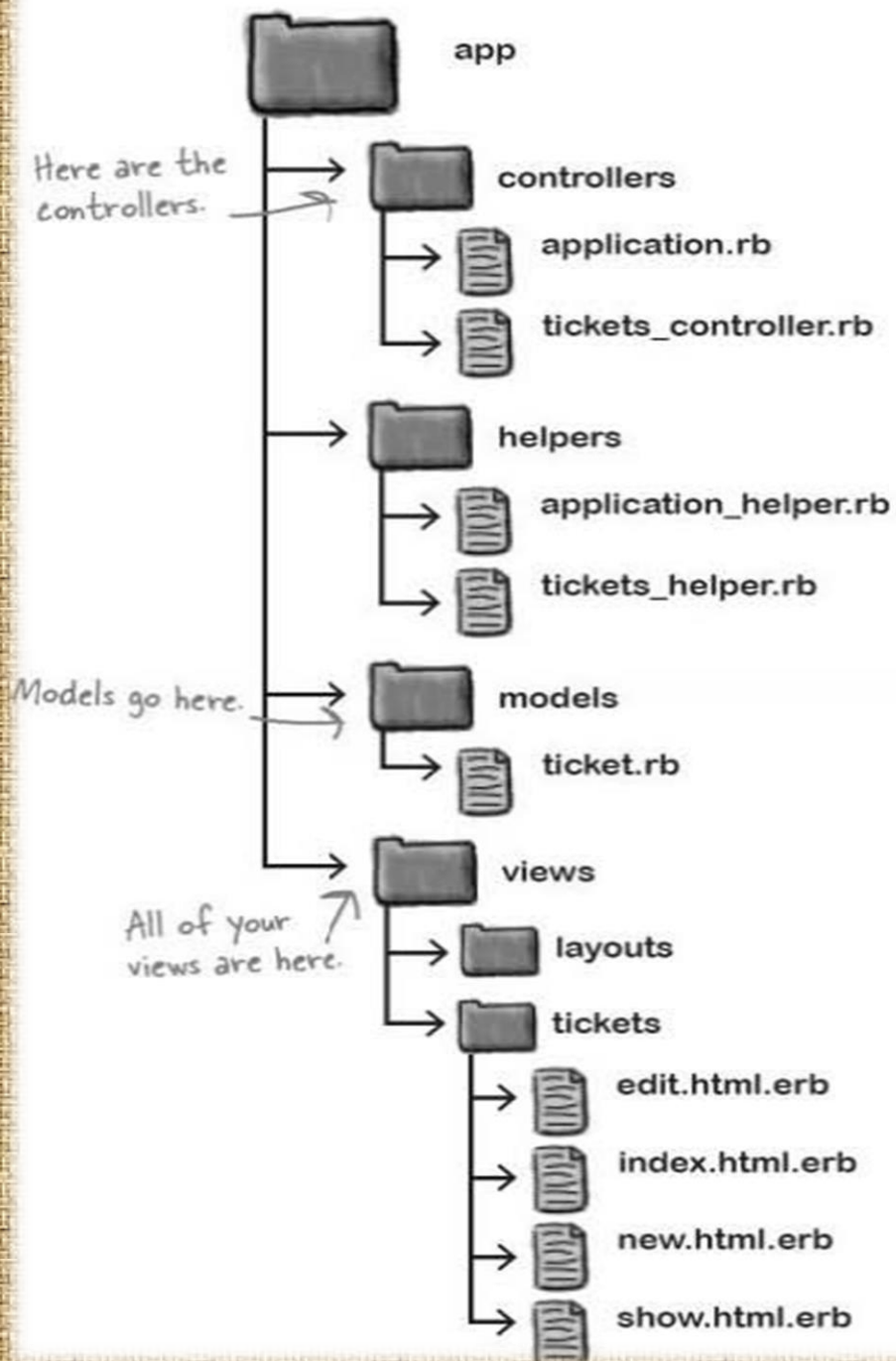
Perguntas ?

- **P:Onde a lógica de negócio deveria ficar na minha aplicação web ?**

R: Tudo vai depender do que você entende por “lógica de negócio”. Algumas pessoas definem como as regras associadas ao gerenciamento de dados. Neste caso, a lógica de negócio fica no modelo. Outros definem como as regras que controlam o fluxo de trabalho do sistema – Neste caso, a lógica de negócio fica no controlador.

- **P:Qual a diferença de controlador e visualização ?**

R: A visualização decide como a aplicação vai parecer e o controlador decide como ela vai funcionar. Sendo assim , a visualização vai definir a cor do botão em uma página web e qual texto aparece na mesma, mas o controlador decidirá o que acontece quando o botão é pressionado.



Alterando arquivos da VIEW

- Se quisermos alterar as descrições e textos nas páginas web, precisamos modificar os códigos que estão em APP/VIEW.
 - Os arquivos de visualização geram as páginas web e são chamados de templates(ou modelos) de página.
-

Então , com o que se parecem os templates (modelos)de página ?

- Abra todos os arquivos **html.erb** na pasta VIEW.
 - Mude as descrições de **seat_id_seq** para **Lugar#**. Altere nos quatro arquivos.
-

FOLDERS

▼ sistemas

▼ app

- ▶ assets
- ▶ controllers
- ▶ helpers
- ▶ mailers
- ▶ models
- ▼ views
 - ▼ layouts
 - ▶ imagens
 - bkn.jpg
 - ▼ sistemas
 - _form.html.erb
 - edit.html.erb
 - index.html.erb
 - new.html.erb
 - show.html.erb

- ▶ config
- ▶ db
- ▶ doc
- ▶ lib
- ▶ log
- ▶ public
- ▶ script
- ▶ test
- ▶ tmp
- ▶ vendor
- .gitignore
- config.ru
- Gemfile
- Gemfile.lock
- Rakefile
- README.rdoc

```
4     <div id="error_explanation">
5       <h2><%= pluralize(@sistema.errors.count, "error") %> prohibited this sistema from being saved:</h2>
6
7       <ul>
8         <% @sistema.errors.full_messages.each do |msg| %>
9           <li><%= msg %></li>
10          <% end %>
11        </ul>
12      </div>
13      <% end %>
14
15      <div class="field">
16        <%= f.label :Nome %><br />
17        <%= f.text_field :name %>
18      </div>
19      <div class="field">
20        <%= f.label :Lugar %><br />
21        <%= f.text_field :seat_id_seq %>
22      </div>
23      <div class="field">
24        <%= f.label :Endereco %><br />
25        <%= f.text_area :address %>
26      </div>
27      <div class="field">
28        <%= f.label :Valor %><br />
29        <%= f.text_field :prprice_paid %>
30      </div>
31      <div class="field">
32        <%= f.label :Email %><br />
33        <%= f.text_field :email_address %>
34      </div>
35      <div class="field">
36        <%= f.label :Phone %><br />
37        <%= f.text_field :phone %>
38      </div>
39      <div class="actions">
40        <%= f.submit %>
41      </div>
42    <% end %>
43
44
```


FOLDERS

▼ sistemas

▼ app

- ▶ assets
- ▶ controllers
- ▶ helpers
- ▶ mailers
- ▶ models
- ▼ views
 - ▼ layouts
 - ▶ Imagens
 - bkn.jpg
 - ▼ sistemas
 - _form.html.erb
 - edit.html.erb
 - index.html.erb
 - new.html.erb
 - show.html.erb

- ▶ config
- ▶ db
- ▶ doc
- ▶ lib
- ▶ log
- ▶ public
- ▶ script
- ▶ test
- ▶ tmp
- ▶ vendor
- .gitignore
- config.ru
- Gemfile
- Gemfile.lock
- Rakefile
- README.rdoc

```
1 <%= render 'form' %>
2 <%= link_to 'Visualizar', @sistema %> |
3 <%= link_to 'Voltar', sistemas_path %>
```

FOLDERS

▼ sistemas

▼ app

▶ assets

▶ controllers

▶ helpers

▶ mailers

▶ models

▼ views

▼ layouts

▶ Imagens

bkn.jpg

▼ sistemas

_form.html.erb

edit.html.erb

index.html.erb

new.html.erb

show.html.erb

▶ config

▶ db

▶ doc

▶ lib

▶ log

▶ public

▶ script

▶ test

▶ tmp

▶ vendor

.gitignore

config.ru

Gemfile

Gemfile.lock

Rakefile

README.rdoc

```

185 <h1>Sistema de Ingressos Online</h1>
186 <h2>Ingressos Cadastro:</h2>
187
188
189
190 <table border=1 cellspacing=0 cellpadding=2 >
191   <tr>
192     <th>Name</th>
193     <th>Lugar</th>
194     <th>Endereco</th>
195     <th>Valor Pago</th>
196     <th>Email</th>
197     <th>Phone</th>
198     <th></th>
199     <th></th>
200     <th></th>
201   </tr>
202
203   <% @sistemas.each do |sistema| %>
204     <tr>
205       <td><%= sistema.name %></td>
206       <td><%= sistema.seat_id_seq %></td>
207       <td><%= sistema.address %></td>
208       <td><%= sistema.prrice_paid %></td>
209       <td><%= sistema.email_address %></td>
210       <td><%= sistema.phone %></td>
211       <td><%= link_to 'Visualizar', sistema %></td>
212       <td><%= link_to 'Editar', edit_sistema_path(sistema) %></td>
213       <td><%= link_to 'Deletar', sistema, method: :delete, data: { confirm: 'Are you sure?' } %></td>
214     </tr>
215   <% end %>
216 </table>
217
218 <br />
219
220 <%= link_to 'Novo Cadastro', new_sistema_path %>
221 </div>
222 </div>
223
224 <div id="footer">&nbsp;</div>
225 </div>
226 </body>

```


Perguntas ?

P: Você chamou `:seat_id_seq` de símbolo. O que é um símbolo ?

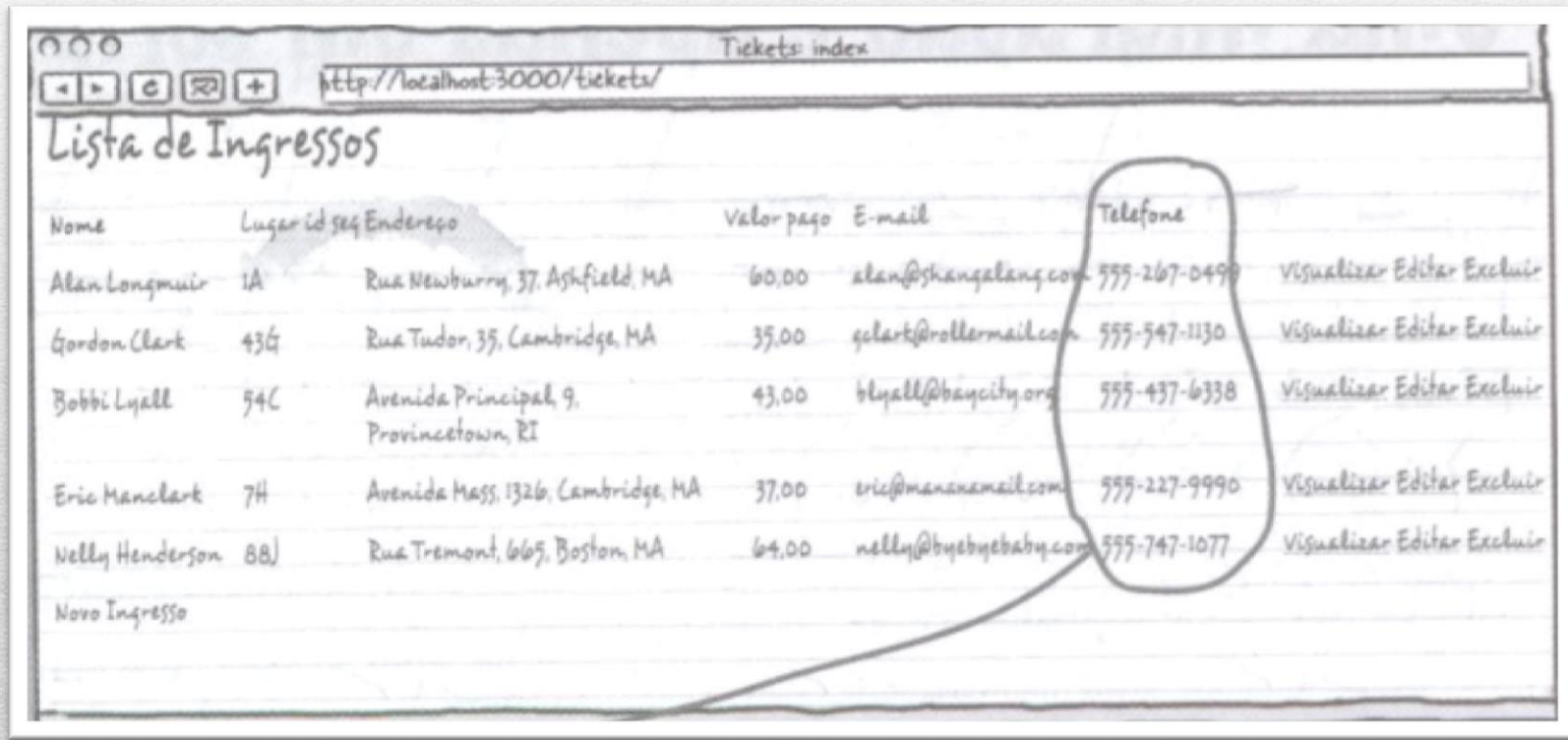
R: Um símbolo é um pouco parecido com uma string. Uma string vem cercada por aspas (“ ”), mas um símbolo sempre é iniciado com dois pontos (:). São geralmente mais usados para nomear coisas no Rails, porque eles são um pouco mais eficientes na memória.

Quero mais uma coisa na
aplicação ! Que os números de
telefones sejam gravados e
exibir na página...



Agora a aplicação precisa armazenar mais informações...

- Precisamos exibir mais um campo em cada página.

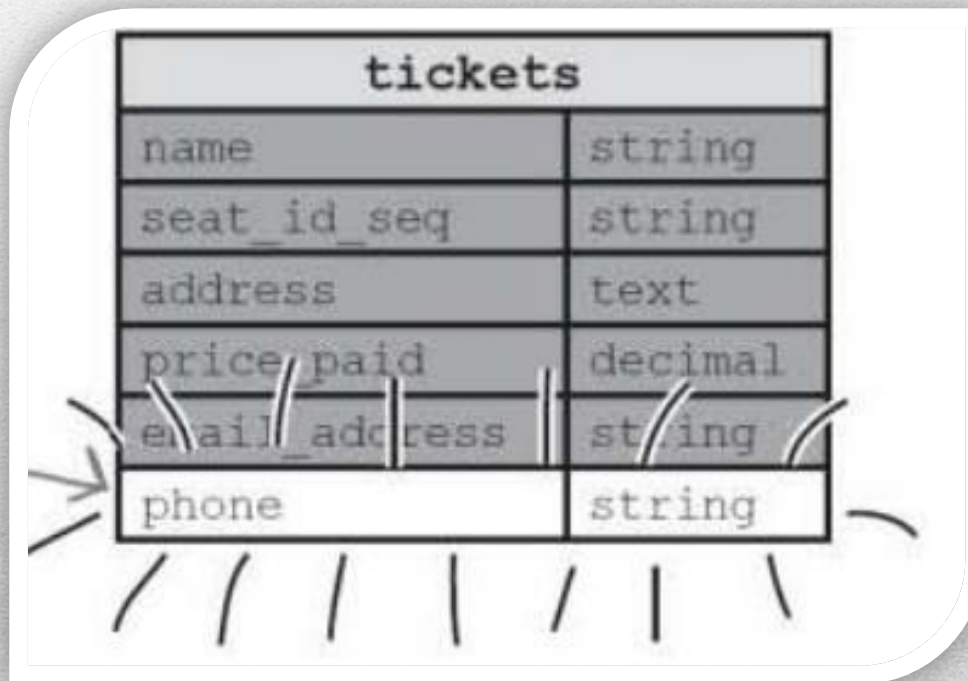


Hand-drawn sketch of a web browser window titled "Tickets: index" with the URL "http://localhost:3000/tickets/". The page displays a table titled "Lista de Ingressos" with the following columns: Nome, Lugar id seq, Endereço, Valor pago, E-mail, Telefone, and actions (Visualizar, Editar, Excluir). The table contains five rows of ticket data. A circle highlights the "Telefone" column, and an arrow points from it to the "Novo Ingresso" button at the bottom left.

Nome	Lugar id seq	Endereço	Valor pago	E-mail	Telefone	Visualizar	Editar	Excluir
Alan Longmuir	1A	Rua Newbury, 37, Ashfield, MA	60,00	alan@shangalang.com	555-267-0499	Visualizar	Editar	Excluir
Gordon Clark	43G	Rua Tudor, 35, Cambridge, MA	35,00	gclark@rollermail.com	555-547-1130	Visualizar	Editar	Excluir
Bobbi Lyall	54C	Avenida Principal, 9, Provincetown, RI	43,00	blyall@baycity.org	555-437-6338	Visualizar	Editar	Excluir
Eric Manclark	7H	Avenida Mass, 1326, Cambridge, MA	37,00	eric@mananemail.com	555-227-9990	Visualizar	Editar	Excluir
Nelly Henderson	88J	Rua Tremont, 665, Boston, MA	64,00	nelly@byebyebaby.com	555-747-1077	Visualizar	Editar	Excluir

Novo Ingresso

- Precisamos armazenar mais uma coluna no bando de dados , para isso vamos usar um script para alterar a estrutura de banco de dados(MIFRAÇÃO).



tickets	
name	string
seat_id_seq	string
address	text
price_paid	decimal
email_address	string
phone	string

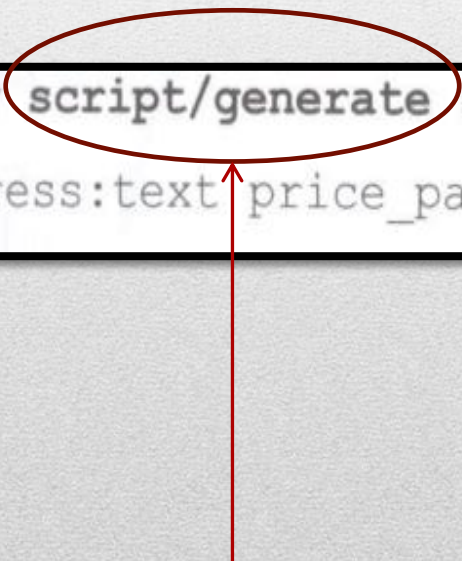
Perguntas ???

- **P:** Alguns códigos na migração parecem que estão recriando a tabela. Por que isso ?
 - **R:** Migrações podem fazer muito mais do que estamos fazendo . Por exemplo ,cada migração tem a habilidade de desfazer-se. É por isso que o código criará uma nova tabela combinada pelo código que pode recriar a tabela.
 - **P:** Se a migração é só um script Ruby , porque tenho que usar o rake ? Por que não posso só executar o script ?
 - **R :** Algumas partes do Ruby são projetadas para serem executadas diretamente e outras não. Migrações são projetadas para serem executadas diretamente.
-

Rails pode gerar migrações

- Quando geramos a aplicação via scaffold utilizamos :

```
ruby script/generate scaffold ticket name:string seat_id_seq:string  
address:text price_paid:decimal email_address:string
```



- ❖ O generate é um script para criar código ruby. É que generate não só escreve código padrão scaffold, mas também gera migrações.
- ❖ Agora suponha que você tenha digitado esse comando :

Ruby generate migration PhoeNumber



NA VERDADE, NÃO DIGITE ISSO !

Motivos ...

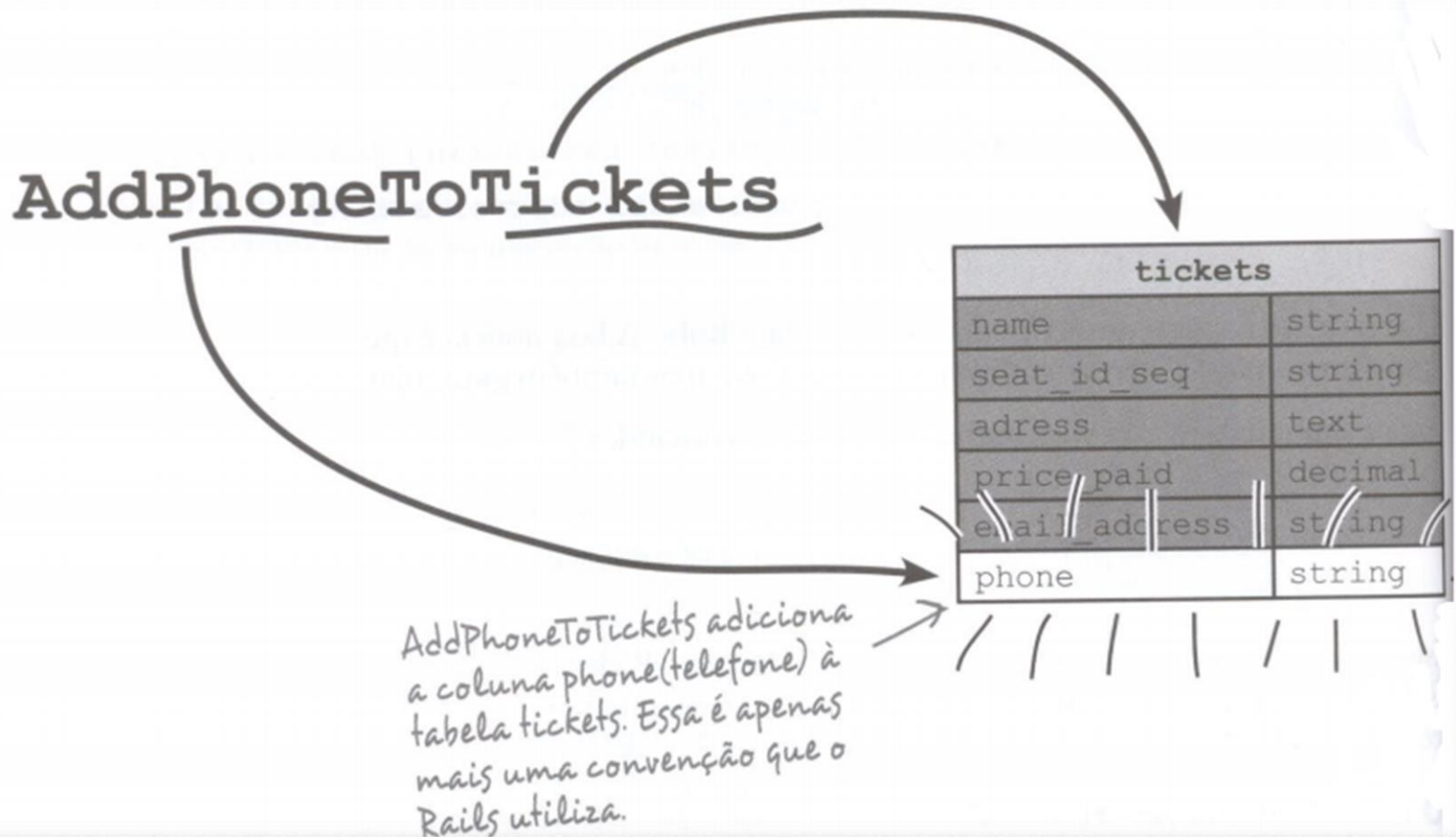
- Geraria um novo arquivo de migração em banco. Poderíamos , então , adicionar o código Ruby para modificar a tabela. O problema é : Até o momento não sabemos como escrever o código para completar o processo de migração.
-

O que podemos fazer ? E o que o nosso bondoso Rails pode fazer por nós ?

- Sabemos que os nomes são realmente importantes no Rails.
- Quando criamos a aplicação com o comando scaffold a chamamos de “tickets” e o Rails deixou gerou uma migração para criar a tabela chamada tickets.
- Lembre-se sempre que convenções em nomes são importantes no Rails e isto poupará o seu trabalho.
- O novo comando para adicionar o campo na tabela será :

Rails generate migration AddPhoneToTickets phone:string

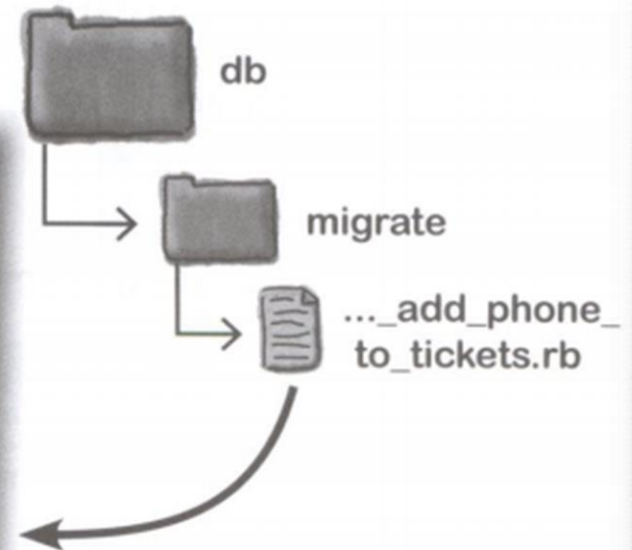
- O Rails sabe que uma migração chamada Add...To... Estará, provavelmente ,adicionando uma nova coluna específica a uma tabela específica, ao invés de apenas gerar uma migração em branco para você preencher, *o Rails vai ,na verdade , escrever para o seu código de migração.*



Agora você precisa executar a sua migração com *rake*.

```
class AddPhoneToTickets < ActiveRecord::Migration
  def self.up
    add_column :tickets, :phone, :string
  end

  def self.down
    remove_column :tickets, :phone
  end
end
```



```
> rake db:migrate
```

Mas alterar só o banco de dados não é suficiente

Editar os modelos (templates)


```
<p>
  <%= f.label :email_address %><br />
  <%= f.text_field :email_address %>
</p>

<p>
  <%= f.label :phone %><br />
  <%= f.text_field :phone %>
</p>

<p>
  <%= f.submit "Update" %>
</p>
```

Isto está dentro do
arquivo edit.html.erb.

Este é o código
adicionado ao arquivo
show.html.erb.



```
<b>Valor pago:</b>  
<%=h @ticket.price_paid %>  
</p>  
  
<p>  
  <b>E-mail:</b>  
  <%=h @ticket.email_address %>  
</p>  
  
<p>
```



```
<p>
  <%= f.label :email_address %><br />
  <%= f.text_field :email_address %>
</p>
<p>
  <%= f.label :phone %><br />
  <%= f.text_field :phone %>
</p>
<p>
```

É um new.html.erb
- a página contendo
o formulário "Criar".

E, finalmente, no arquivo index.html.erb, que gera a lista de todos os ingressos. Precisamos adicionar código em dois lugares aqui: no cabeçalho da coluna e em cada linha da tabela.

```
<th>E-mail</th>
  <th>Telefone</th>
</tr>
<% for ticket in @tickets %>
  <tr>
    <td><%=h ticket.name %></td>
    <td><%=h ticket.seat_id_seq %></td>
    <td><%=h ticket.address %></td>
    <td><%=h ticket.price_paid %></td>
    <td><%=h ticket.email_address %></td>
    <td><%=h ticket.phone %></td>
    <td><%= link_to 'Exibir', ticket %></td>
```


- Altere o modelo : models/tickets/tickets.rb
Adicione mais um parâmetro :

:phone

```
1  class Sistema < ActiveRecord::Base
2    attr_accessible :name , :phone
3  end
4
```

Perguntas ...

- **P:** Por que tem `<%=h ... %>` em alguns lugares? O que significa “h” ?
 - **R:** “h” é um método auxiliar(ou Helper). Auxiliares são utilizados para coisas como formatar saída. O auxiliar “h” escapa caracteres especiais no campo como “<” e “&”. Isto evitará que pessoas enviem texto para sites que contenham JavaScript ou outro tipo de código.
 - **P :** Por que as strings são utilizadas em alguns lugares e os Símbolos em outros?
 - **R:** Strings são utilizados em modelos (templates) de página onde uma parte simples de texto é necessária. Símbolos (as palavras que começam com “ : ”)são, comumente ,utilizadas em descrições / Títulos de campos.
-

