

LOCABUILDERS

LevelDev 4







Ruby on Rails

Um framework HTTP/Web [on steroids]

CRUD (on Rails)





Edição

routes.rb

```
get "/topics", to: "topics#index"
post "/topics", to: "topics#create"
get "/topics/new", to: "topics#new", as: :new_topic

get "/topics/:id", to: "topics#show", as: :topic
```



routes.rb

```
get "/topics", to: "topics#index"
post "/topics", to: "topics#create"
get "/topics/new", to: "topics#new", as: :new_topic
get "/topics/:id/edit", to: "topics#edit", as: :edit_topic
get "/topics/:id", to: "topics#show", as: :topic
```



routes.rb

```
get "/topics", to: "topics#index"
post "/topics", to: "topics#create"
get "/topics/new", to: "topics#new", as: :new_topic
get "/topics/:id/edit", to: "topics#edit", as: :edit_topic
get "/topics/:id", to: "topics#show", as: :topic
```

app/views/topics/show.html.erb

```
<div>
  <%= link_to "Topics", topics_path %> |
</div>
```



routes.rb

```
get "/topics", to: "topics#index"
post "/topics", to: "topics#create"
get "/topics/new", to: "topics#new", as: :new_topic
get "/topics/:id/edit", to: "topics#edit", as: :edit_topic
get "/topics/:id", to: "topics#show", as: :topic
```

app/views/topics/show.html.erb

```
<div>
  <%= link_to "Topics", topics_path %> |
  <%= link_to "Editar" %>
</div>
```

routes.rb

```
get "/topics", to: "topics#index"
post "/topics", to: "topics#create"
get "/topics/new", to: "topics#new", as: :new_topic
get "/topics/:id/edit", to: "topics#edit", as: :edit_topic
get "/topics/:id", to: "topics#show", as: :topic
```

app/views/topics/show.html.erb

```
<div>
  <%= link_to "Topics", topics_path %> |
  <%= link_to "Editar", edit_topic_path(@topic) %>
</div>
```

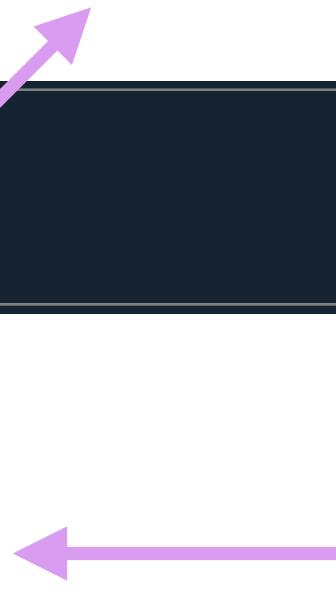


routes.rb

```
get "/topics", to: "topics#index"
post "/topics", to: "topics#create"
get "/topics/new", to: "topics#new", as: :new_topic
get "/topics/:id/edit", to: "topics#edit", as: :edit_topic
get "/topics/:id", to: "topics#show", as: :topic
```

app/views/topics/show.html.erb

```
<div>
  <%= link_to "Topics", topics_path %> |
  <%= link_to "Editar", edit_topic_path(@topic) %>
</div>
```



routes.rb

```
get "/topics", to: "topics#index"
post "/topics", to: "topics#create"
get "/topics/new", to: "topics#new", as: :new_topic
get "/topics/:id/edit", to: "topics#edit", as: :edit_topic
get "/topics/:id", to: "topics#show", as: :topic
```

app/controllers/topics_controller.rb

```
class TopicsController < ApplicationController
  def show
    @topic = Topic.find(params[:id])
  end

  def edit
  end
end
```



hello.rb

```
get "/topics", to: "topics#index"
post "/topics", to: "topics#create"
get "/topics/new", to: "topics#new", as: :new_topic
get "/topics/:id/edit", to: "topics#edit", as: :edit_topic
get "/topics/:id", to: "topics#show", as: :topic
```

app/controllers/topics_controller.rb

```
class TopicsController < ApplicationController
  def show
    @topic = Topic.find(params[:id])
  end

  def edit
    @topic = Topic.find(params[:id])
  end
end
```



app/controllers/topics_controller.rb

```
print "hello world"

class TopicsController < ApplicationController
  def show
    @topic = Topic.find(params[:id])
  end

  def edit
    @topic = Topic.find(params[:id])
  end
end
```



app/controllers/topics_controller.rb

```
print "hello world"

class TopicsController < ApplicationController
  def show
    @topic = Topic.find(params[:id])
  end

  def edit
    @topic = Topic.find(params[:id])
  end

  def set_topic
    @topic = Topic.find(params[:id])
  end
end
```



app/controllers/topics_controller.rb

```
print "hello world"

class TopicsController < ApplicationController
  def show
    @topic = Topic.find(params[:id])
  end

  def edit
    set_topic
  end

  def set_topic
    @topic = Topic.find(params[:id])
  end
end
```



app/controllers/topics_controller.rb

```
print "hello world"

class TopicsController < ApplicationController
  def show
    set_topic
  end

  def edit
    set_topic
  end

  def set_topic
    @topic = Topic.find(params[:id])
  end
end
```



```
print "hello world"

class TopicsController < ApplicationController
  def show
    set_topic
  end

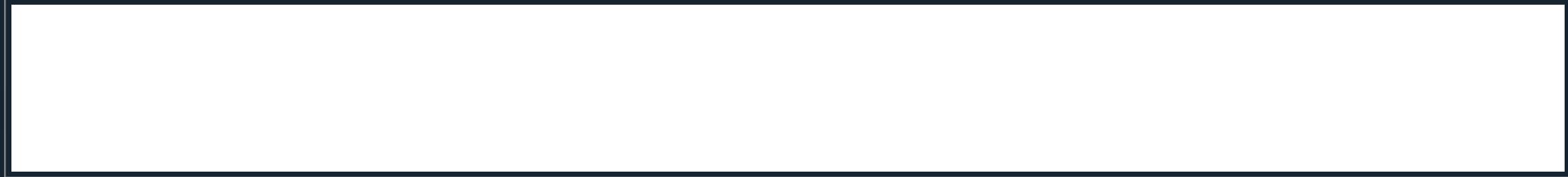
  def edit
    set_topic
  end

  private

  def set_topic
    @topic = Topic.find(params[:id])
  end
end
```



app/views/topics/edit.html.erb



app/views/topics/edit.html.erb

```
<%= form_with(model: @topic) do |form| %>
```



app/views/topics/edit.html.erb

```
<%= form_with(model: @topic) do |form| %>
<% if @topic.errors.any? %>
```



app/views/topics/edit.html.erb

```
<%= form_with(model: @topic) do |form| %>
  <% if @topic.errors.any? %>
    <div style="color: red">
```



app/views/topics/edit.html.erb

```
<%= form_with(model: @topic) do |form| %>
  <% if @topic.errors.any? %>
    <div style="color: red">
```



app/views/topics/edit.html.erb

```
<%= render "form", topic: @topic %>
```

app/views/topics/new.html.erb

```
<%= form_with(model: @topic) do |form| %>
```

```
  <% if @topic.errors.any? %>
    <div style="color: red">
      <ul>
        <% @topic.errors.each do |error| %>
          <li><%= error.full_message %></li>
        <% end %>
      </ul>
    </div>
  <% end %>

  <div>
    <%= form.label :titulo, style: "display: block" %>
    <%= form.text_field :titulo %>
  </div>

  <div>
    <%= form.submit %>
  </div>
```

```
<% end %>
```

```
<div>
  <%= link_to "topics", topics_path %>
</div>
```

app/views/topics/edit.html.erb

```
<%= render "form", topic: @topic %>
```

app/views/topics/new.html.erb

```
<%= form_with(model: @topic) do |form| %>
```

```
  <% if @topic.errors.any? %>
    <div style="color: red">
      <ul>
        <% @topic.errors.each do |error| %>
          <li><%= error.full_message %></li>
        <% end %>
      </ul>
    </div>
  <% end %>
```

```
  <div>
    <%= form.label :titulo, style: "display: block" %>
    <%= form.text_field :titulo %>
  </div>
```

```
  <div>
    <%= form.submit %>
  </div>
```

```
<% end %>
```

```
<div>
  <%= link_to "topics", topics_path %>
</div>
```

}

app/views/topics/edit.html.erb

```
<%= render "form", topic: @topic %>
```

app/views/topics/new.html.erb

```
<%= form_with(model: @topic) do |form| %>
<% if @topic.errors.any? %>
<div style="color: red">
<ul>
<% @topic.errors.each do |error| %>
<li><%= error.full_message %></li>
<% end %>
</ul>
</div>
<% end %>
<div>
<%= form.label :titulo, style: "display: block" %>
<%= form.text_field :titulo %>
</div>
<div>
<%= form.submit %>
</div>
<% end %>
```

```
<div>
<%= link_to "topics", topics_path %>
</div>
```



topics/_form.html.erb

app/views/topics/_form.html.erb

```
<%= form_with(model: @topic) do |form| %>
  <% if @topic.errors.any? %>
    <div style="color: red">
      <ul>
        <% @topic.errors.each do |error| %>
          .....
```



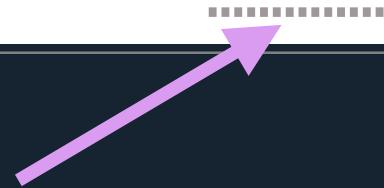
app/views/topics/_form.html.erb

```
<%= form_with(model: topic) do |form| %>
  <% if @topic.errors.any? %>
    <div style="color: red">
      <ul>
        <% @topic.errors.each do |error| %>
          .....
```



app/views/topics/_form.html.erb

```
<%= form_with(model: topic) do |form| %>
  <% if topic.errors.any? %>
    <div style="color: red">
      <ul>
        <% @topic.errors.each do |error| %>
```



app/views/topics/_form.html.erb

```
<%= form_with(model: topic) do |form| %>
  <% if topic.errors.any? %>
    <div style="color: red">
      <ul>
        <% topic.errors.each do |error| %>
          .....
        </ul>
    </div>
  <% end %>
<= form_with(model: topic)>
```



app/views/topics/edit.html.erb

```
<%= render "form" %>
```



app/views/topics/edit.html.erb

```
<%= render "form", topic: @topic %>
```



app/views/topics/edit.html.erb

```
<%= render "form", topic: @topic %>
```

app/views/topics/new.html.erb

```
<%= render "form", topic: @topic %>

<div>
  <%= link_to "topics", topics_path %>
</div>
```



browser



<http://localhost:3000/topics/1/edit>

```
Started PATCH "/topicos/9" for ::1 at 2023-03-01 12:59:20 -0300
```

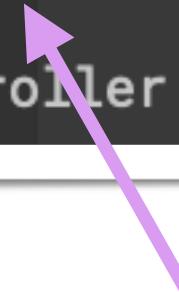
```
ActionController::RoutingError (No route matches [PATCH] "/topicos/9"):
```

browser



<http://localhost:3000/topics/1/edit>

```
Started PATCH "/topicos/9" for ::1 at 2023-03-01 12:59:20 -0300  
ActionController::RoutingError (No route matches [PATCH] "/topicos/9"):
```



browser



<http://localhost:3000/topics/1/edit>

```
Started PATCH "/topicos/9" for ::1 at 2023-03-01 12:59:20 -0300  
ActionController::RoutingError (No route matches [PATCH] "/topicos/9"):
```



<http://localhost:3000/topics/1/edit>

browser

```
Started PATCH "/topicos/9" for ::1 at 2023-03-01 12:59:20 -0300  
ActionController::RoutingError (No route matches [PATCH] "/topicos/9"):
```



<http://localhost:3000/topics/1/edit>

browser

hello.rb

```
get "/topics", to: "topics#index"
post "/topics", to: "topics#create"
get "/topics/new", to: "topics#new", as: :new_topic
get "/topics/:id/edit", to: "topics#edit", as: :edit_topic
get "/topics/:id", to: "topics#show", as: :topic
```



hello.rb

```
get "/topics", to: "topics#index"
post "/topics", to: "topics#create"
get "/topics/new", to: "topics#new", as: :new_topic
get "/topics/:id/edit", to: "topics#edit", as: :edit_topic
get "/topics/:id", to: "topics#show", as: :topic
patch "/topics/:id", to: "topics#update"
```



hello.rb

```
get "/topics", to: "topics#index"
post "/topics", to: "topics#create"
get "/topics/new", to: "topics#new", as: :new_topic
get "/topics/:id/edit", to: "topics#edit", as: :edit_topic
get "/topics/:id", to: "topics#show", as: :topic
patch "/topics/:id", to: "topics#update"
put "/topics/:id", to: "topics#update"
```



app/controllers/topics_controller.rb

```
class TopicsController
  def update
    set_topic
    topic_params = params.require(:topic).permit(:titulo)

    if @topic.update(topic_params)
      redirect_to topic_url(@topic)
    else
      render :edit, status: :unprocessable_entity
    end
  end
end
```





Action Hooks!

Isolando comportamentos repetidos

app/controllers/topics_controller.rb

```
class TopicsController < ApplicationController

  def show
    set_topic
  end

  def edit
    set_topic
  end
end
```



app/controllers/topics_controller.rb

```
class TopicsController < ApplicationController
  before_action :set_topic, only: [ :show, :edit ]

  def show
    set_topic
  end

  def edit
    set_topic
  end
end
```



app/controllers/topics_controller.rb

```
class TopicsController < ApplicationController
  before_action :set_topic, only: [ :show, :edit ]

  def show
    end

  def edit
    end
end
```



app/controllers/topics_controller.rb

```
class TopicsController < ApplicationController
  before_action :set_topic, only: [:show, :edit]

  def show
  end

  def edit
  end
end
```



app/controllers/topics_controller.rb

```
class TopicsController < ApplicationController
  before_action :set_topic, only: [:show, :edit]

  def show
  end

  def edit
  end
end
```



app/controllers/topics_controller.rb

```
class TopicsController < ApplicationController
  before_action :set_topic, only: [:show, :edit]

  def update
    set_topic
    topic_params = params.require(:topic).permit(:titulo)

    if @topic.update(topic_params)
      redirect_to topic_url(@topic)
    else
      render :new, status: :unprocessable_entity
    end
  end
end
```

app/controllers/topics_controller.rb

```
class TopicsController < ApplicationController
  before_action :set_topic, only: [:show, :edit]

  def update
    set_topic
    topic_params = params.require(:topic).permit(:titulo)

    if @topic.update(topic_params)
      redirect_to topic_url(@topic)
    else
      render :new, status: :unprocessable_entity
    end
  end
end
```



app/controllers/topics_controller.rb

```
class TopicsController < ApplicationController
  before_action :set_topic, only: [:show, :edit, :update]

  def update
    set_topic
    topic_params = params.require(:topic).permit(:titulo)

    if @topic.update(topic_params)
      redirect_to topic_url(@topic)
    else
      render :new, status: :unprocessable_entity
    end
  end
end
```



app/controllers/topics_controller.rb

```
class TopicsController < ApplicationController
  before_action :set_topic, only: [:show, :edit, :update]

  def update
    topic_params = params.require(:topic).permit(:titulo)

    if @topic.update(topic_params)
      redirect_to topic_url(@topic)
    else
      render :new, status: :unprocessable_entity
    end
  end
end
```





Mais limpeza!
Isolando comportamentos: boa prática!

app/controllers/topics_controller.rb

```
class TopicsController < ApplicationController
  before_action :set_topic, only: [:show, :edit, :update]

  def update
    topic_params = params.require(:topic).permit(:titulo)
    if @topic.update(topic_params)
      ...
    end

  def create
    topic_params = params.require(:topic).permit(:titulo)
    @topic = Topic.new(topic_params)
    ...
  end
end
```

app/controllers/topics_controller.rb

```
class TopicsController < ApplicationController
  before_action :set_topic, only: [:show, :edit, :update]

  def update
    topic_params = params.require(:topic).permit(:titulo)
    if @topic.update(topic_params)
      ...
    end
  end

  def create
    topic_params = params.require(:topic).permit(:titulo)
    @topic = Topic.new(topic_params)
    ...
  end
end
```



app/controllers/topics_controller.rb

```
class TopicsController < ApplicationController
  before_action :set_topic, only: [:show, :edit, :update]

  def update
    topic_params = params.require(:topic).permit(:titulo)

    if @topic.update(topic_params)
      ...
    end
  end

  def create
    topic_params = params.require(:topic).permit(:titulo)
    @topic = Topic.new(topic_params)

    ...
  end
end
```



app/controllers/topics_controller.rb

```
class TopicsController < ApplicationController
  before_action :set_topic, only: [:show, :edit, :update]

  def update
    if @topic.update(topic_params)
      ...
    end
  end

  def create
    @topic = Topic.new(topic_params)
    ...
  end
end
```



app/controllers/topics_controller.rb

```
class TopicsController < ApplicationController
  before_action :set_topic, only: [:show, :edit, :update]

  def update
    if @topic.update(topic_params)
      ...
    end
  end

  private

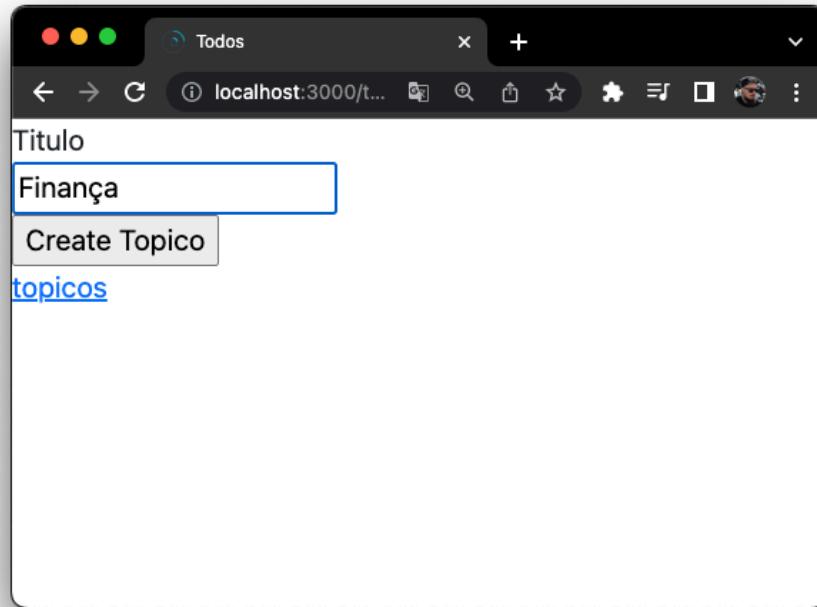
  def topic_params
    params.require(:topic).permit(:titulo)
  end
end
```

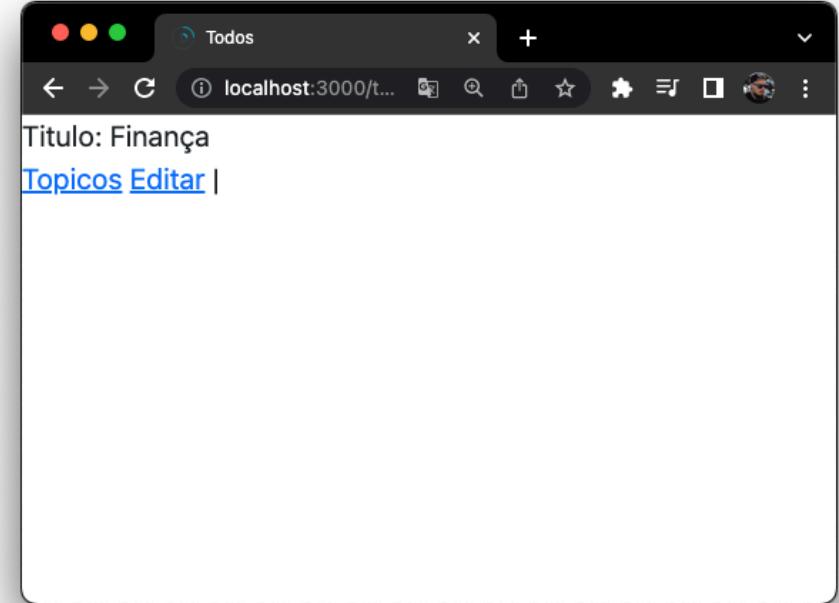
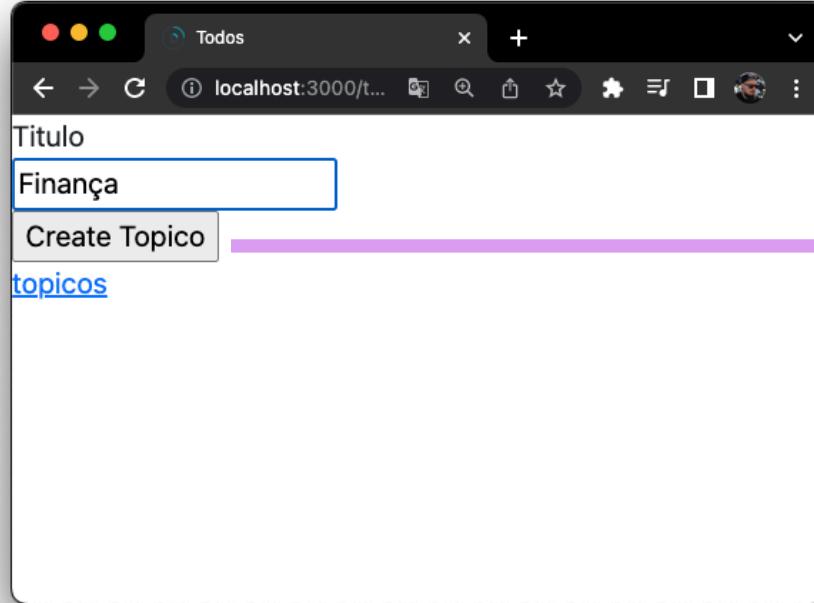


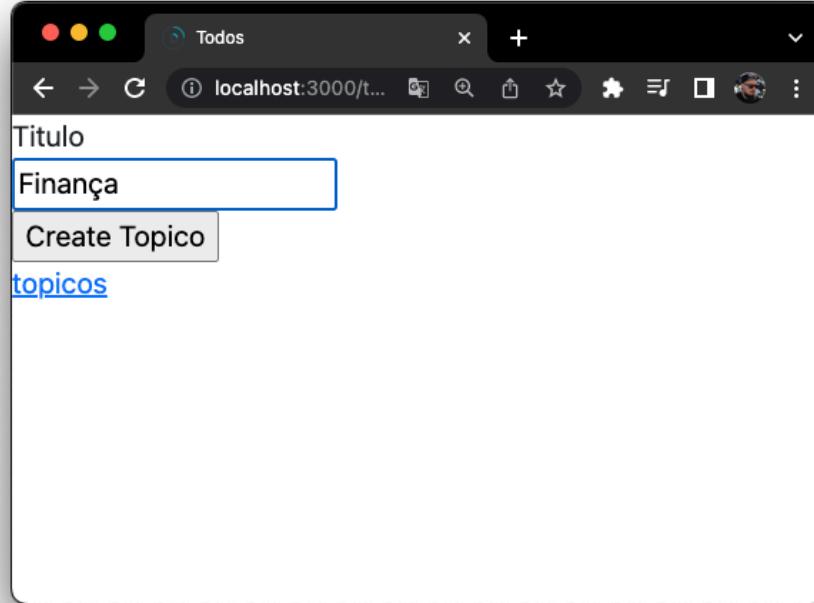


Tratando o Sucesso?

Comunicação com o cliente é chave.

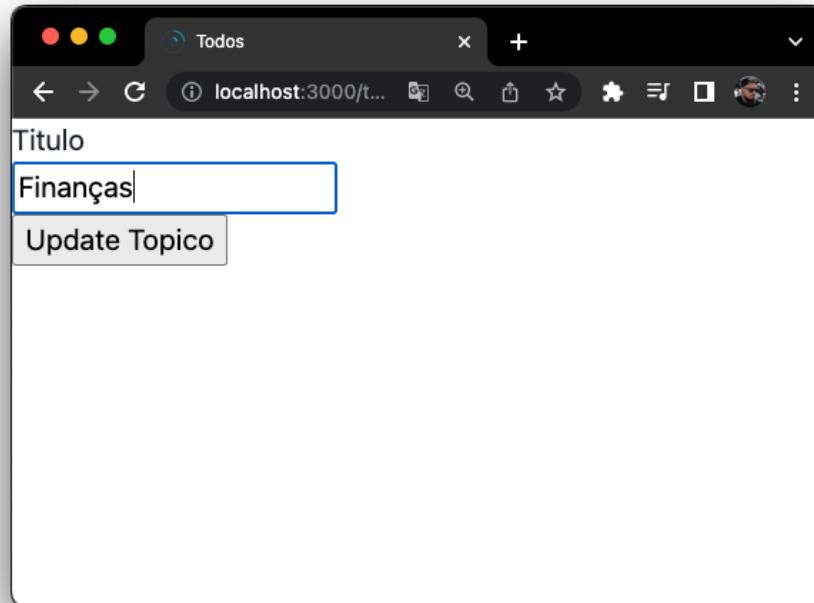






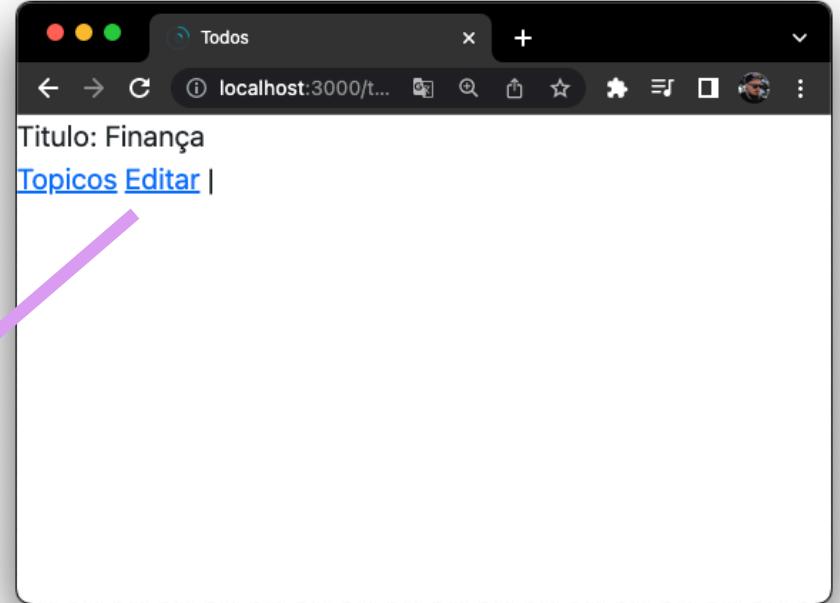
Título

Create Topico

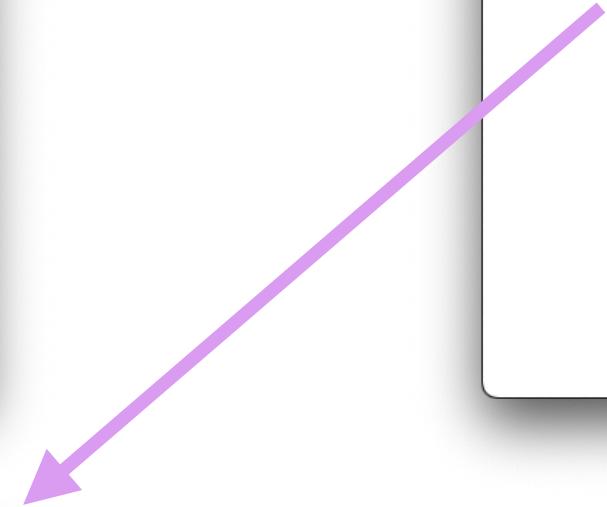


Título

Update Topico



Título: Finança
[Topicos](#) [Editar](#) |



Título

Create Topico

[topicos](#)

Título: Finança

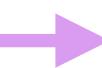
[Topicos Editar](#) |

Título

Update Topico

Título: Finanças

[Topicos Editar](#) |



app/controllers/topics_controller.rb

```
def update
  if @topic.update(topic_params)
    redirect_to topic_url(@topic)
  else
    render :new, status: :unprocessable_entity
  end
end
```



app/controllers/topics_controller.rb

```
def update
  if @topic.update(topic_params)
    flash[:notice]
    redirect_to topic_url(@topic)
  else
    render :new, status: :unprocessable_entity
  end
end
```



app/controllers/topics_controller.rb

```
def update
  if @topic.update(topic_params)
    flash[:notice] = "Tópico atualizado" ←
    redirect_to topic_url(@topic)
  else
    render :new, status: :unprocessable_entity
  end
end
```



app/controllers/topics_controller.rb

```
def update
  if @topic.update(topic_params)
    flash[:notice] = "Tópico atualizado"
    redirect_to topic_url(@topic)
  else
    render :new, status: :unprocessable_entity
  end
end
```

app/views/topics/show.html.erb

```
<p style="color: green"><%= notice %></p>
```

.....

app/controllers/topics_controller.rb

```
def update
  if @topic.update(topic_params)
    flash[:notice] = "Tópico atualizado"
    redirect_to topic_url(@topic)
  else
    render :new, status: :unprocessable_entity
  end
end
```

app/views/topics/show.html.erb

```
<p style="color: green"><%= notice %></p>
```

.....



é exibido no topo da página

app/controllers/topics_controller.rb

```
def update
  if @topic.update(topic_params)
    flash[:notice] = "Tópico atualizado"
    redirect_to topic_url(@topic)
  else
    render :new, status: :unprocessable_entity
  end
end
```

app/views/topics/show.html.erb

```
<p style="color: green"><%= notice %></p>
```

.....

app/controllers/topics_controller.rb

```
def update
  if @topic.update(topic_params)
    flash[:notice] = "Tópico atualizado"
    redirect_to topic_url(@topic)
  else
    render :new, status: :unprocessable_entity
  end
end
```

app/views/topics/show.html.erb

```
<% if !notice.nil? && notice != "" %>
  <p style="color: green"><%= notice %></p>
<% end %>
....
```



app/controllers/topics_controller.rb

```
def update
  if @topic.update(topic_params)
    flash[:notice] = "Tópico atualizado"
    redirect_to topic_url(@topic)
  else
    render :new, status: :unprocessable_entity
  end
end
```

app/views/topics/show.html.erb

```
<% if notice.present? %>
  <p style="color: green"><%= notice %></p>
<% end %>
....
```



app/controllers/topics_controller.rb

```
def update
  if @topic.update(topic_params)
    redirect_to topic_url(@topic)
  else
    render :new, status: :unprocessable_entity
  end
end
```



app/controllers/topics_controller.rb

```
def update
  if @topic.update(topic_params)
    redirect_to topic_url(@topic), notice: "Tópico atualizado"
  else
    render :new, status: :unprocessable_entity
  end
end
```



app/controllers/topics_controller.rb

```
def update
  if @topic.update(topic_params)
    redirect_to topic_url(@topic), notice: "Tópico atualizado"
  else
    render :new, status: :unprocessable_entity
  end
end

def create
  @topic = Topic.new(topic_params)

  if @topic.save
    redirect_to topic_url(@topic)
  else
    render :new, status: :unprocessable_entity
  end
end
```

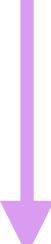


app/controllers/topics_controller.rb

```
def update
  if @topic.update(topic_params)
    redirect_to topic_url(@topic), notice: "Tópico atualizado"
  else
    render :new, status: :unprocessable_entity
  end
end

def create
  @topic = Topic.new(topic_params)

  if @topic.save
    redirect_to topic_url(@topic), notice: "Tópico criado"
  else
    render :new, status: :unprocessable_entity
  end
end
```





Fechando o CRUD

Excluindo registros “do jeitão rails”.

app/views/topics/show.html.erb

```
<div>
  <%= link_to "Topics", topics_path %> |
  <%= link_to "Editar", edit_topic_path(@topic) %>

</div>
```



app/views/topics/show.html.erb

```
<div>
  <%= link_to "Topics", topics_path %> |
  <%= link_to "Editar", edit_topic_path(@topic) %>

  <%= button_to "Remover tópico", @topic %>
</div>
```



app/views/topics/show.html.erb

```
<div>
  <%= link_to "Topics", topics_path %> |
  <%= link_to "Editar", edit_topic_path(@topic) %>

  <%= button_to "Remover tópico", @topic %>
</div>
```

```
<form class="button_to" method="post" action="/topics/15">
  <input type="hidden" name="_method" value="patch" autocomplete="off" />
  <button type="submit">Destroy this task</button>
  <input type="hidden" name="authenticity_token" value="npYLH..." autocomplete="off" />
</form>
```

app/views/topics/show.html.erb

```
<div>
  <%= link_to "Topics", topics_path %> |
  <%= link_to "Editar", edit_topic_path(@topic) %>

  <%= button_to "Remover tópico", @topic %>
</div>
```



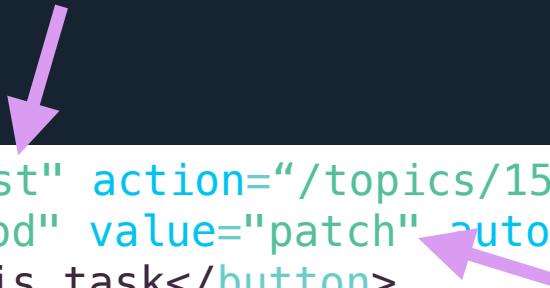
```
<form class="button_to" method="post" action="/topics/15">
  <input type="hidden" name="_method" value="patch" autocomplete="off" />
  <button type="submit">Destroy this task</button>
  <input type="hidden" name="authenticity_token" value="npYLH..." autocomplete="off" />
</form>
```

app/views/topics/show.html.erb

```
<div>
  <%= link_to "Topics", topics_path %> |
  <%= link_to "Editar", edit_topic_path(@topic) %>

  <%= button_to "Remover tópico", @topic %>
</div>
```

```
<form class="button_to" method="post" action="/topics/15">
  <input type="hidden" name="_method" value="patch" autocomplete="off" />
  <button type="submit">Destroy this task</button>
  <input type="hidden" name="authenticity_token" value="npYLH..." autocomplete="off" />
</form>
```



app/views/topics/show.html.erb

```
<div>
  <%= link_to "Topics", topics_path %> |
  <%= link_to "Editar", edit_topic_path(@topic) %>

  <%= button_to "Remover tópico", @topic %>
</div>
```



app/views/topics/show.html.erb

```
<div>
  <%= link_to "Topics", topics_path %> |
  <%= link_to "Editar", edit_topic_path(@topic) %>

  <%= button_to "Remover tópico", @topic, method: :delete %>
</div>
```



app/controllers/topics_controller.rb

```
def destroy  
  @topic.destroy  
  
  redirect_to topics_url  
end
```



app/controllers/topics_controller.rb

```
def destroy
  @topic.destroy

  redirect_to topics_url, notice: "Tópico removido."
end
```



app/controllers/topics_controller.rb

```
before_action :set_topic, only: [:show, :edit, :update, :destroy]

def destroy
  @topic.destroy

  redirect_to topics_url, notice: "Tópico removido."
end
```



app/controllers/topics_controller.rb

```
before_action :set_topic, only: [:show, :edit, :update, :destroy]  
  
def destroy  
  @topic.destroy  
  
  redirect_to topics_url, notice: "Tópico removido."  
end
```



app/controllers/topics_controller.rb

```
before_action :set_topic, only: [:show, :edit, :update, :destroy]  
  
def destroy  
  @topic.destroy  
  redirect_to topics_url, notice: "Tópico removido."  
end
```



app/controllers/topics_controller.rb

```
def destroy
  @topic.destroy

  redirect_to topics_url, notice: "Tópico removido."
end
```

app/views/topics/index.html.erb

```
<p style="color: green"><%= notice %></p>
```



app/controllers/topics_controller.rb

```
def destroy
  @topic.destroy

  redirect_to topics_url, notice: "Tópico removido."
end
```

app/views/topics/index.html.erb

```
<% if notice.present? %>
  <p style="color: green"><%= notice %></p>
<% end %>
```



app/controllers/topics_controller.rb

```
def destroy
  @topic.destroy

  redirect_to topics_url, notice: "Tópico removido."
end
```

app/views/topics/index.html.erb

```
<% if notice.present? %>
  <p style="color: green"><%= notice %></p>
<% end %>
```

config/routes.rb

```
delete "/topics/:id", to: "topics#destroy"
```



Extra: scaffolding

[o default para ser sobreescrito]

Terminal

```
$ rails g scaffold post title:string body:text  
$ rails db:migrate
```



```
root@9d90190c0229:/usr/src/peeper/app# rails g scaffold post title:string body:text
```

```
- 3.2.0 ➜ ricardovaleriano@juquery ➤ 2023-07/app ➡ main
```

```
find lib-1.7.4, lib-T.7.4 in locally installed gems  
e install` to install missing gems.
```

```
0.0 ➜ 3.2.0 ➜ ricardovaleriano@juquery ➤ 2023-07/app ➡ main
```

```
- 3.2.0 ➜ ricardovaleriano@juquery ➤ 2023-07/app ➡ main
```

```
- 3.2.0 ➜ ricardovaleriano@juquery ➤ 2023-07/app ➡ main
```

```
- 3.2.0 ➜ ricardovaleriano@juquery ➤ 2023-07/app ➡ main
```

```
| pygmentize -f rtf -l yaml -O fontface=menlo,style=paraiso-light | pbcopy  
lexer for alias 'yaml' found
```

```
- 3.2.0 ➜ ricardovaleriano@juquery ➤ 2023-07/app ➡ main
```

```
| pygmentize -f rtf -l yaml -O fontface=menlo,style=paraiso-light | pbcopy
```

```
- 3.2.0 ➜ ricardovaleriano@juquery ➤ 2023-07/app ➡ main
```

```
| pygmentize -f rtf -l yaml -O fontface=menlo,style=paraiso-light | pbcopy
```

```
- 3.2.0 ➜ ricardovaleriano@juquery ➤ 2023-07/app ➡ main
```

```
drop
```

```
find lib-1.7.4, lib-T.7.4 in locally installed gems  
e install` to install missing gems.
```

db/migrate/.....create_posts.rb

```
class CreatePosts < ActiveRecord::Migration[7.0]
  def change
    create_table :posts do |t|
      t.string :title
      t.text :body

      t.timestamps
    end
  end
end
```



db/migrate/.....create_posts.rb

```
class CreatePosts < ActiveRecord::Migration[7.0]
  def change
    create_table :posts do |t| ← Tabela
      t.string :title
      t.text :body

      t.timestamps
    end
  end
end
```



db/migrate/.....create_posts.rb

```
class CreatePosts < ActiveRecord::Migration[7.0]
  def change
    create_table :posts do |t|
      t.string :title
      t.text :body
      t.timestamps
    end
  end
end
```

Annotations:

- A purple arrow points from the word "Tabela" to the "create_table" method.
- A purple arrow points from the word "Colunas" to the "t.string" and "t.text" lines.

db/migrate/.....create_posts.rb

```
class CreatePosts < ActiveRecord::Migration[7.0]
  def change
    create_table :posts do |t|
      t.string :title
      t.text :body
      t.timestamps
    end
  end
end
```

Annotations on the code:

- A purple arrow points from the text "Tabela" to the line `create_table :posts do |t|`.
- A purple arrow points from the text "Colunas" to the line `t.string :title`.
- A purple arrow points from the text "Colunas ‘do Rails’" to the line `t.timestamps`.

Tabelas

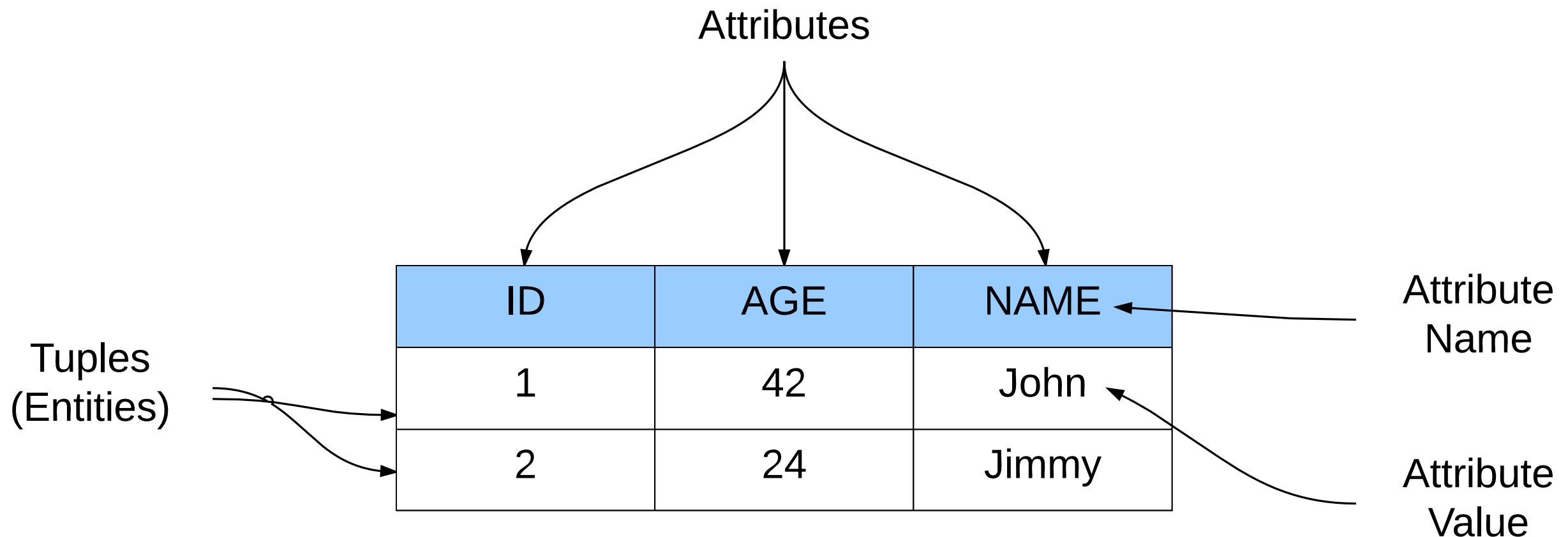
[as entidades que vivem em um modelo]

Colunas

[características e propriedades]

Linhas (tuplas)

[informação guiada por colunas]





[ActiveRecord (oo)]

Post .count
.where

post.title = ""
.body = ""

post.save
.delete

[Relacional]

Tabelas

Colunas

Linhas (tuplas)

Chaves [keys]



```
$ rails dbconsole
SQLite version 3.39.5 2022-10-14 20:58:05
Enter ".help" for usage hints.
sqlite> .schema posts
CREATE TABLE IF NOT EXISTS "posts" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL,
    "title" varchar, "body" text, "created_at" datetime(6) NOT NULL, "updated_at" datet
ime(6) NOT NULL);
sqlite> █
```



_rails_command dbconsole

⌘x2

```
$ rails dbconsole
SQLite version 3.39.5 2022-10-14 20:58:05
Enter ".help" for usage hints.
sqlite> .schema posts
CREATE TABLE IF NOT EXISTS "posts" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL,
 "title" varchar, "body" text, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL);
sqlite> █
```



_rails_command dbconsole

⌘x2

```
$ rails dbconsole
SQLite version 3.39.5 2022-10-14 20:58:05
Enter ".help" for usage hints.
sqlite> .schema posts
CREATE TABLE IF NOT EXISTS "posts" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL,
    "title" varchar, "body" text, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL);
sqlite> █
```



Primárias

[identificação única de linhas (tuplas)]



FINANÇAS PESSOAIS

Dá mesmo para ter um RG diferente em cada Estado?

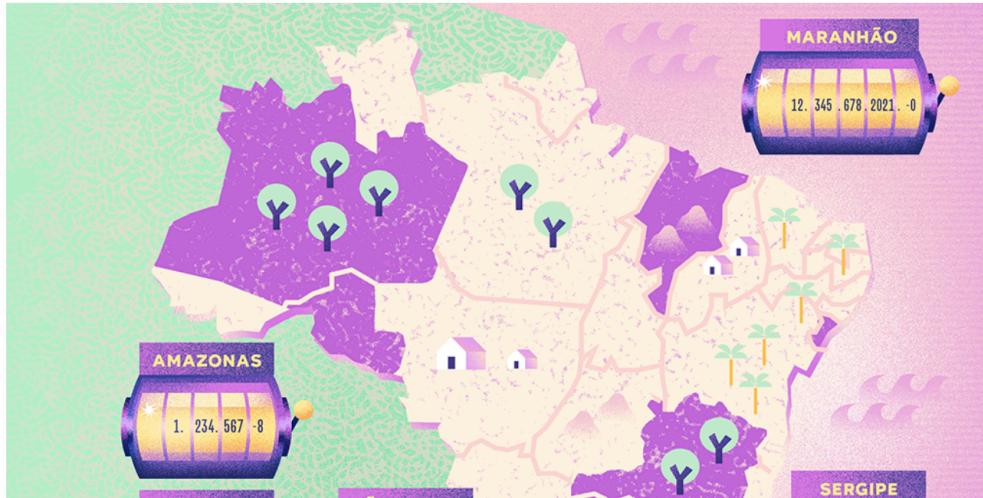
É possível ter 27 RGs diferentes, de fato – amostra de que o documento se tornou um fóssil burocrático.

Por **Juliana Américo**

Atualizado em 16 jun 2021, 13h57 - Publicado em 10 Maio 2021, 15h38

CONTINUA APÓS PUBLICIDADE

APRESENTADO POR COLÉGIO
BANDEIRANTES

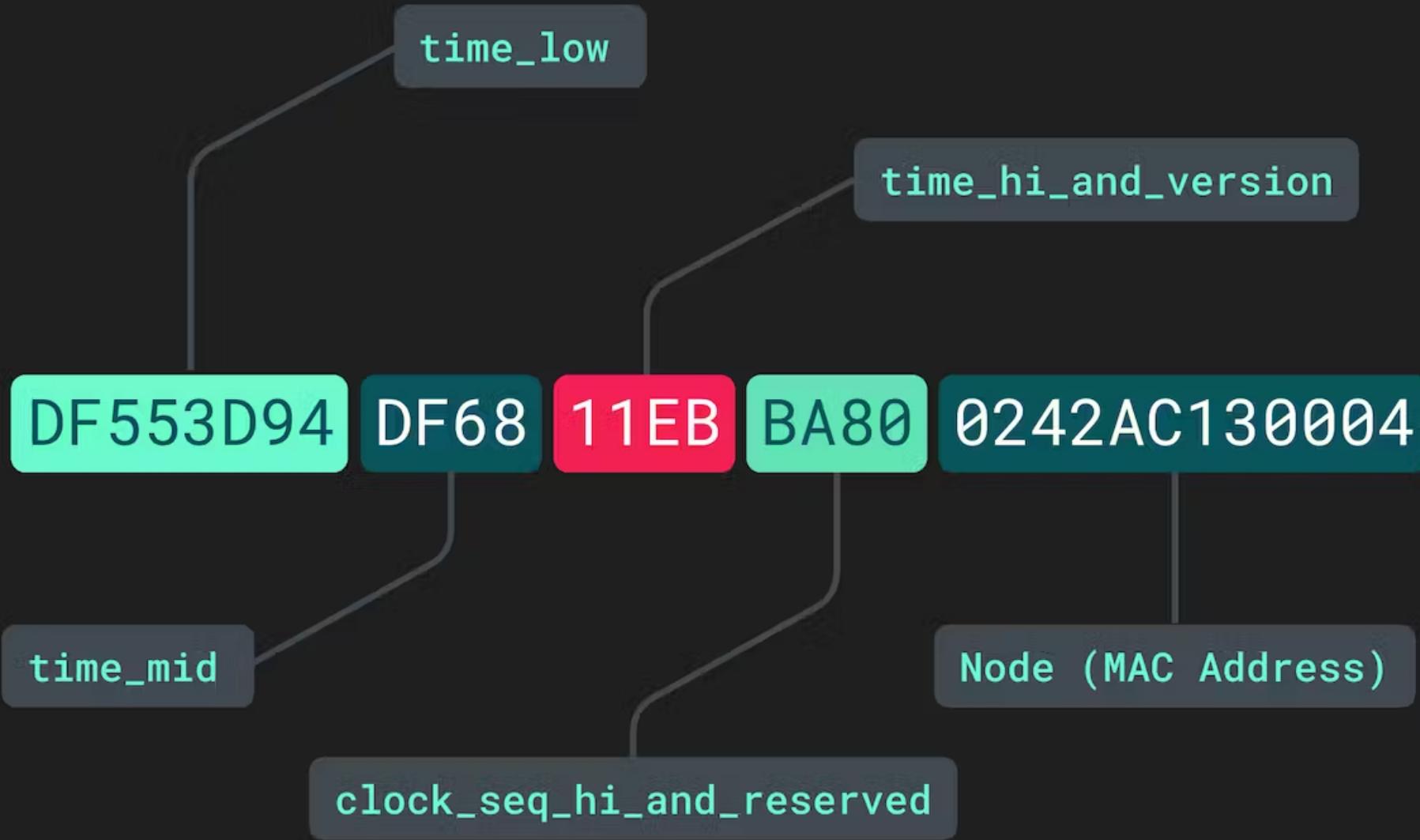


Crie conteúdo de e-mail melhor hoje

Cadastre-se



UUID





Vantagens do UUID como chave primária?

- Privacidade (?): “impossível” chutar um ID
- Não “localidade”: sem sincronização entre servidores
- Pode ser gerado em qualquer lugar (eg.: Frontend)



Active Record and PostgreSQL X +

guides.rubyonrails.org/v5.0/active_record_postgresql.html

Home Guides Index ▾ Contribute Credits

Active Record and PostgreSQL

This guide covers PostgreSQL specific usage of Active Record.

After reading this guide, you will know:

- ✓ How to use PostgreSQL's datatypes.
- ✓ How to use UUID primary keys.
- ✓ How to implement full text search with PostgreSQL.
- ✓ How to back your Active Record models with database views.

In order to use the PostgreSQL adapter you need to have at least version 9.1 installed. Older versions are not supported.

To get started with PostgreSQL have a look at the [configuring Rails guide](#). It describes how to properly setup Active Record for PostgreSQL.

1 Datatypes

PostgreSQL offers a number of specific datatypes. Following is a list of types, that are supported by

Chapters

- 1. [Datatypes](#)
 - [Bytea](#)
 - [Array](#)
 - [Hstore](#)
 - [JSON](#)
 - [Range Types](#)
 - [Composite Types](#)
 - [Enumerated Types](#)
 - [UUID](#)
 - [Bit String Types](#)
 - [Network Address Types](#)
 - [Geometric Types](#)
- 2. [UUID Primary Keys](#)
- 3. [Full Text Search](#)
- 4. [Database Views](#)

Active Record and PostgreSQL X +

guides.rubyonrails.org/v5.0/active_record_postgresql.html

Home Guides Index ▾ Contribute Credits

Active Record and PostgreSQL

This guide covers PostgreSQL specific usage of Active Record.

After reading this guide, you will know:

- ✓ How to use PostgreSQL's datatypes.
- ✓ How to use UUID primary keys.
- ✓ How to implement full text search with PostgreSQL.
- ✓ How to back your Active Record models with database views.

In order to use the PostgreSQL adapter you need to have at least version 9.1 installed. Older versions are not supported.

To get started with PostgreSQL have a look at the [configuring Rails guide](#). It describes how to properly setup Active Record for PostgreSQL.

1 Datatypes

PostgreSQL offers a number of specific datatypes. Following is a list of types, that are supported by

Chapters

- 1. [Datatypes](#)
 - [Bytea](#)
 - [Array](#)
 - [Hstore](#)
 - [JSON](#)
 - [Range Types](#)
 - [Composite Types](#)
 - [Enumerated Types](#)
 - [UUID](#)
 - [Bit String Types](#)
 - [Network Address Types](#)
 - [Geometric Types](#)
- 2. [UUID Primary Keys](#)
- 3. [Full Text Search](#)
- 4. [Database Views](#)



ActiveRecord::AttributeMethod X +

← → C api.rubyonrails.org/classes/ActiveRecord/AttributeMetho... 🔍 ⌂ ☆ Update : Link

Search (/) for a class, method, ...

▶ files

▶ Core extensions

▶ AbstractController

▶ ActionCable

▶ ActionController

▶ ActionDispatch

▶ ActionMailbox

▶ ActionMailer

▶ ActionText

▶ ActionView

▶ ActiveJob

▶ ActiveModel

▼ ActiveRecord

 ActiveJobRequiredError < ActiveRecord::Act

 ActiveRecordError < StandardError

 AdapterNotFound < ActiveRecord::ActiveRe

 AdapterNotSpecified < ActiveRecord::Active

 AdapterTimeout < ActiveRecord::QueryAbor

▶ Aggregations

 AssociationTypeMismatch < ActiveRecord::A

▶ Associations

 AsynchronousQueryInsideTransactionError <

primary_key()

Defines the primary key field – can be overridden in subclasses. Overwriting will negate any effect of the `primary_key_prefix_type` setting, though.

Source: [show](#) | [on GitHub](#)

primary_key=(value)

Link

Sets the name of the primary key column.

```
class Project < ActiveRecord::Base
  self.primary_key = 'sysid'
end
```

You can also define the `primary_key` method yourself:

```
class Project < ActiveRecord::Base
  def self.primary_key
    'foo_' + super
  end
end

Project.primary_key # => "foo_id"
```

Source: [show](#) | [on GitHub](#)

quoted_primary_key()

Link

Returns a quoted version of the primary key name, used to construct SQL statements.

Source: [show](#) | [on GitHub](#)

[fictício] unico_demais.rb

```
class UnicoDemais < ApplicationRecord  
  
end
```



[fictício] unico_demais.rb

```
class UnicoDemais < ApplicationRecord  
  primary_key = :uuid  
  
end
```



[fictício] unico_demais.rb

```
class UnicoDemais < ApplicationRecord  
  primary_key = :uuid  
  
  def set_uuid  
    self.uuid = SecureRandom.uuid  
  end  
end
```



[fictício] unico_demais.rb

```
class UnicoDemais < ApplicationRecord
  before_create :set_uuid
  primary_key = :uuid

  def set_uuid
    self.uuid = SecureRandom.uuid
  end
end
```



Bônus

Escapando ao console do SQLite: .quit



_rails_command dbconsole

⌘2

```
$ rails dbconsole
SQLite version 3.39.5 2022-10-14 20:58:05
Enter ".help" for usage hints.
sqlite> .schema posts
CREATE TABLE IF NOT EXISTS "posts" ("id" integer PRIMARY KEY AUTOINCREMENT NOT NULL,
    "title" varchar, "body" text, "created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL);
sqlite> █
```

Terminal

```
$ rails g model comment post:references body:text  
$ rails db:migrate
```



```
$ rails g model comment post:references body:text
  invoke  active_record
  create    db/migrate/20230831134238_create_comments.rb
  create    app/models/comment.rb
  invoke  test_unit
  create    test/models/comment_test.rb
  create    test/fixtures/comments.yml
```

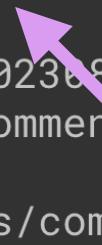
⌚3.10.0 ➤ ⚙3.2.0 ➤ ricardovaleriano@juquery ➤ Mentoria-2023_06/blogy ➤ ↗ main

```
$ rails db:migrate
== 20230831134238 CreateComments: migrating =====
-- create_table(:comments)
-> 0.0062s
== 20230831134238 CreateComments: migrated (0.0063s) =====
```

⌚3.10.0 ➤ ⚙3.2.0 ➤ ricardovaleriano@juquery ➤ Mentoria-2023_06/blogy ➤ ↗ main

```
$
```

```
$ rails g model comment post:references body:text
  invoke  active_record
  create    db/migrate/20230831134238_create_comments.rb
  create    app/models/comment.rb
  invoke  test_unit
  create    test/models/comment_test.rb
  create    test/fixtures/comments.yml
```



```
⌚3.10.0 ➤ ⚙3.2.0 ➤ ricardovaleriano@juquery ➤ Mentoria-2023_06/blogy ➤ ↗ main
```

```
$ rails db:migrate
== 20230831134238 CreateComments: migrating =====
-- create_table(:comments)
 -> 0.0062s
== 20230831134238 CreateComments: migrated (0.0063s) =====
```

```
⌚3.10.0 ➤ ⚙3.2.0 ➤ ricardovaleriano@juquery ➤ Mentoria-2023_06/blogy ➤ ↗ main
```

```
$
```

db/schema.rb

```
create_table "comments", force: :cascade do |t|
  t.integer "post_id", null: false
  t.text "body"
  t.datetime "created_at", null: false
  t.datetime "updated_at", null: false
  t.index ["post_id"], name: "index_comments_on_post_id"
end
```



db/schema.rb

```
create_table "comments", force: :cascade do |t|
  t.integer "post_id", null: false ←
  t.text "body"
  t.datetime "created_at", null: false
  t.datetime "updated_at", null: false
  t.index ["post_id"], name: "index_comments_on_post_id"
end
```



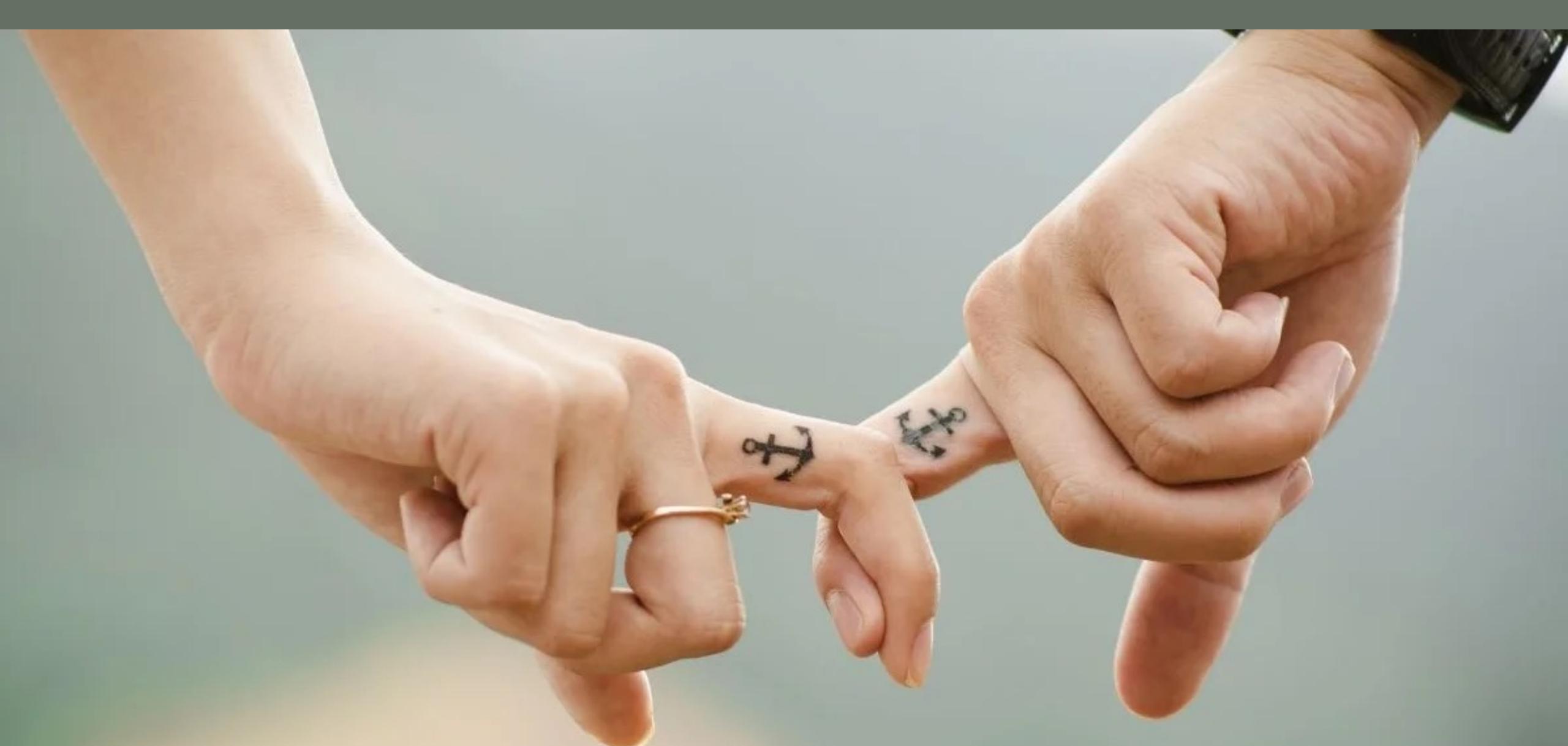
db/schema.rb

```
create_table "comments", force: :cascade do |t|
  t.integer "post_id", null: false ←
  t.text "body"
  t.datetime "created_at", null: false
  t.datetime "updated_at", null: false
  t.index ["post_id"], name: "index_comments_on_post_id"
end
```



Chaves Estrangeiras

[relação (link) entre tabelas]



Integridade Referencial



Bônus

Faker: uma gem para gerar dados “aleatórios”

GitHub - faker-ruby/faker: A lib

github.com/faker-ruby/faker

Update :

README.md



Faker

Faker

Tests passing gem version 3.2.1 docs test coverage 98% maintainability ?

Faker is a port of [Perl's Data::Faker library](#). It's a library for generating fake data such as names, addresses, and phone numbers.

Faker helps you generate realistic test data, and populate your database with more than a couple of records while you're doing development.

It comes in very handy for taking screenshots (taking screenshots for a personal project) and it was the original impetus for the creation of this gem).

+ 835 contributors

Languages

Ruby 100.0%



```
irb(main):006> █
```

```
< 7.0.7.2
```

```
2.11.5
```

```
ord 7.0.7.2
```

```
.0
```

```
le 7.0.7.2
```

```
rage 7.0.7.2
```

```
ler 7.0.7.2
```

```
7.0.7.2
```

```
-rails 3.4.2
```

```
lbox 7.0.7.2
```

```
-rails 1.2.1
```

```
c 7.0.7.2
```

```
rails 1.2.2
```

```
le 4.2.0
```

```
ls 1.4.0
```

```
.7.2
```

```
orm_notifier 1.16.0
```

```
7.0.7
```

```
et 7.0.7
```

```
18 Gemfile dependencies, 79 gems now installed.
```

```
o [gemname] to see where a bundled gem is installed.
```

```
2.0 ➤ ricardovaleriano@juquery ➤ Mentoria-2023_06/blogy ➤ main ➤
```

```
ails g bullet:install
```

```
in config/environments/development.rb
```

```
to enable bullet in test environment? (y/n) y
```

```
in config/environments/test.rb
```

```
2.0 ➤ ricardovaleriano@juquery ➤ Mentoria-2023_06/blogy ➤ main ➤
```



```
2.0 ➤ ricardovaleriano@juquery ➤ Mentoria-2023_06/blogy ➤ main ➤
```

Gemfile

```
group :development do
  gem "web-console"
  gem "faker"
end
```

```
$ bundle install
$ rails c
```

