Curso de Ruby on Rails – Aulas 1 a 19
Jackson Pires

https://www.youtube.com/watch?v=ZHPondVB9RQ&list=PLe3LRfCs4go-mkvHRMSXEOG-HDbzesyaP

Resumo do curso feito por Roberto Pinheiro

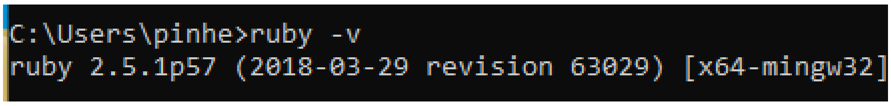
Instalando o Ruby e o Rails

- Instale o Ruby v. 2.5.1.2 com msys2 64bit e Rails v. 5.2.1

- Baixar o Ruby em:

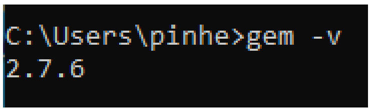
https://rubyinstaller.org/downloads/archives/

ruby -v



```
C:\Users\pinhe>ruby -v
ruby 2.5.1p57 (2018-03-29 revision 63029) [x64-mingw32]
```

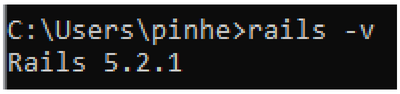
gem -v



```
C:\Users\pinhe>gem -v
2.7.6
```

gem install rails -v 5.2.1

rails -v



```
C:\Users\pinhe>rails -v
Rails 5.2.1
```

Aula 05 – Documentação Ruby, IRB e Pry

Documentação

Documentação oficial

https://ruby-doc.org/

Documentação paralela colaborativa

https://apidock.com/ruby

Onde aprender?

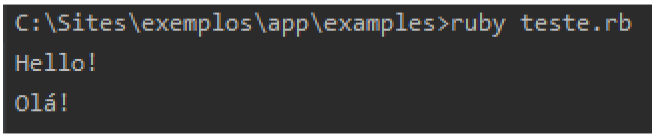
http://tryruby.org

https://www.jmonteiro.com/aprendaaprogramar/

Como interpretar um arquivo Ruby?

app/examples/teste.rb

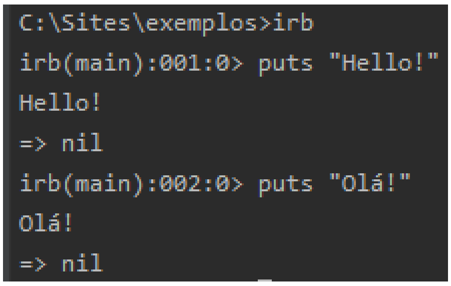
```
puts "Hello!"
puts "Olá!"
```



```
C:\Sites\exemplos\app\examples>ruby teste.rb
Hello!
Olá!
```

irb (Interactive Ruby Shell)

O irb interpreta código Ruby.



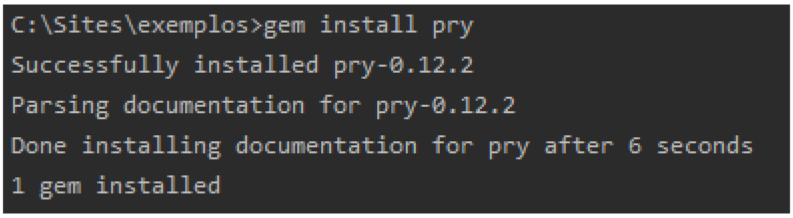
```
C:\Sites\exemplos>irb
irb(main):001:0> puts "Hello!"
Hello!
=> nil
irb(main):002:0> puts "Olá!"
Olá!
=> nil
```

Pry

É uma alternativa ao irb.

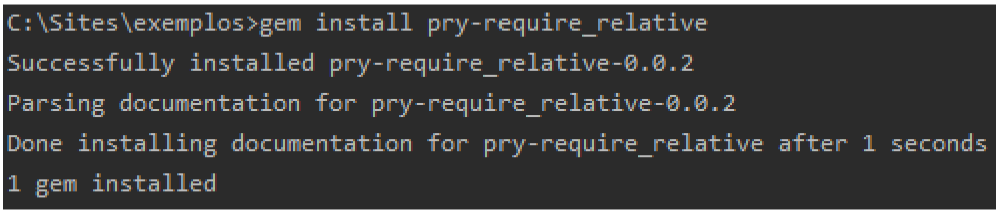
Para instalar, entre no terminal com o comando:

gem install pry

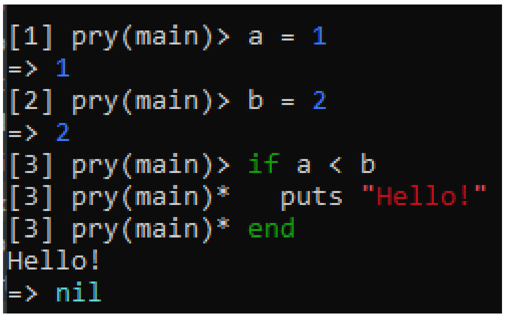


```
C:\Sites\exemplos>gem install pry
Successfully installed pry-0.12.2
Parsing documentation for pry-0.12.2
Done installing documentation for pry after 6 seconds
1 gem installed
```

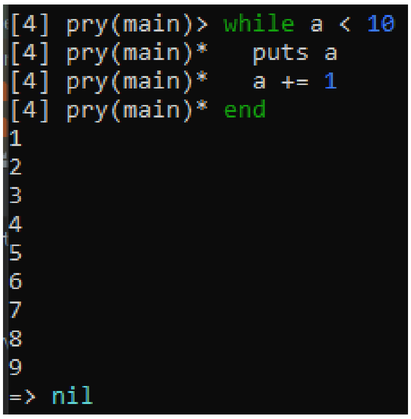
gem install pry-require_relative



```
C:\Sites\exemplos>gem install pry-require_relative
Successfully installed pry-require_relative-0.0.2
Parsing documentation for pry-require_relative-0.0.2
Done installing documentation for pry-require_relative after 1 seconds
1 gem installed
```



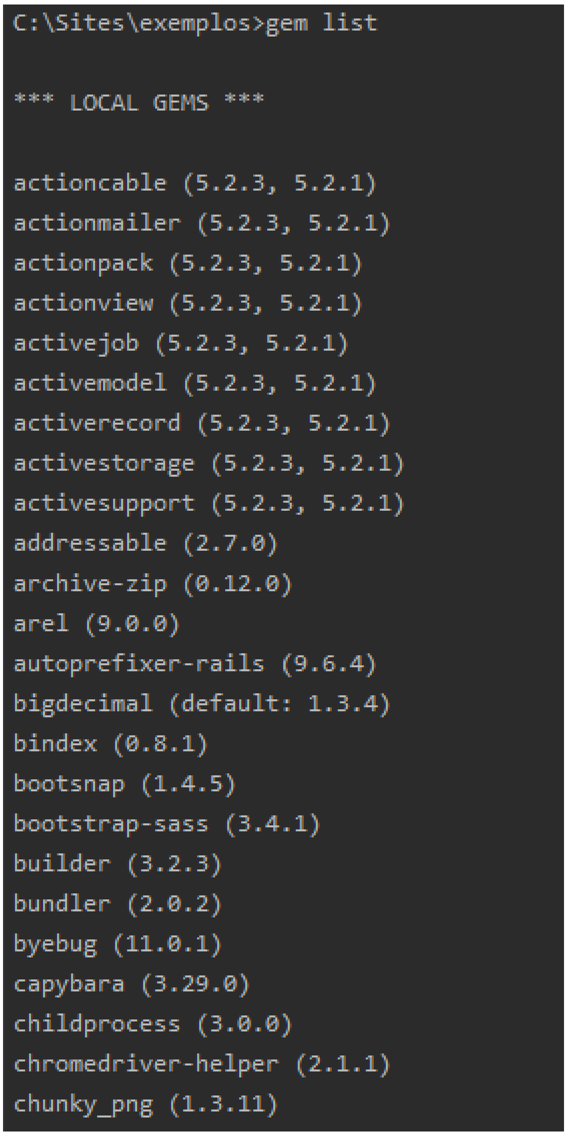
```
[1] pry(main)> a = 1
=> 1
[2] pry(main)> b = 2
=> 2
[3] pry(main)> if a < b
[3] pry(main)*   puts "Hello!"
[3] pry(main)* end
Hello!
=> nil
```



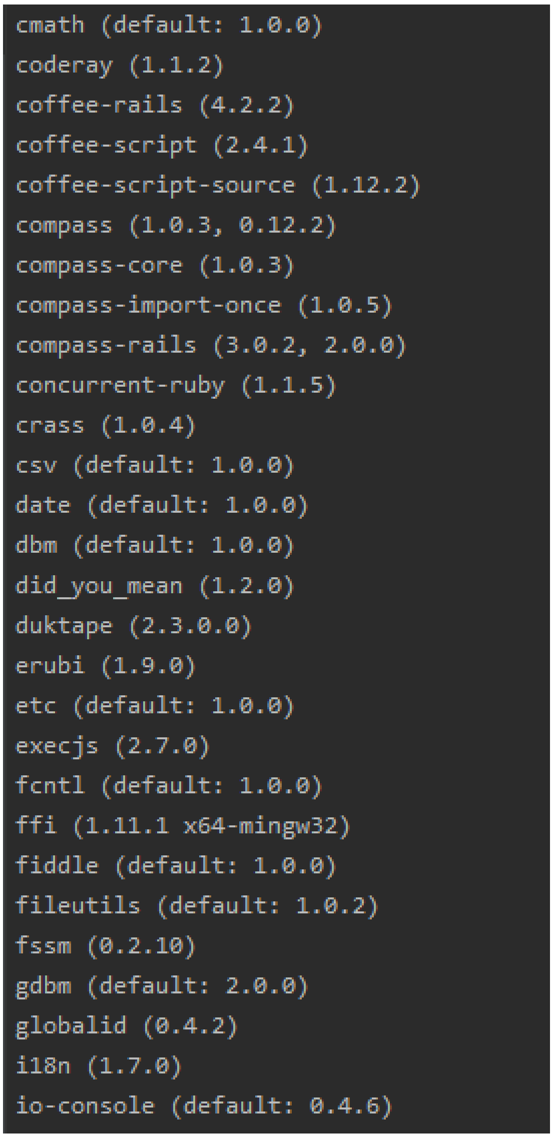
```
[4] pry(main)> while a < 10
[4] pry(main)*   puts a
[4] pry(main)*   a += 1
[4] pry(main)* end
1
2
3
4
5
6
7
8
9
=> nil
```

Listando as gems

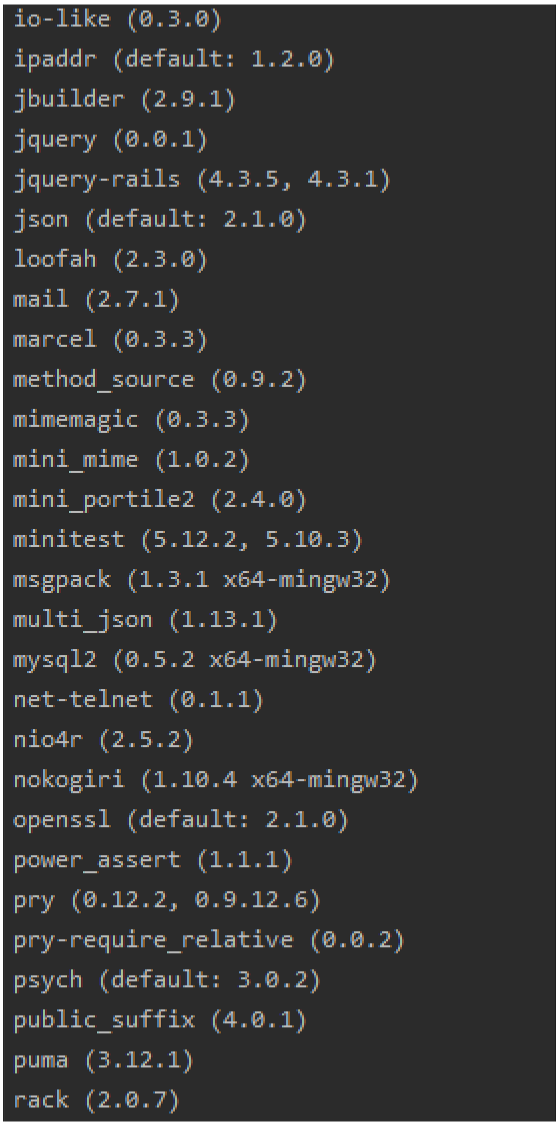
gem list



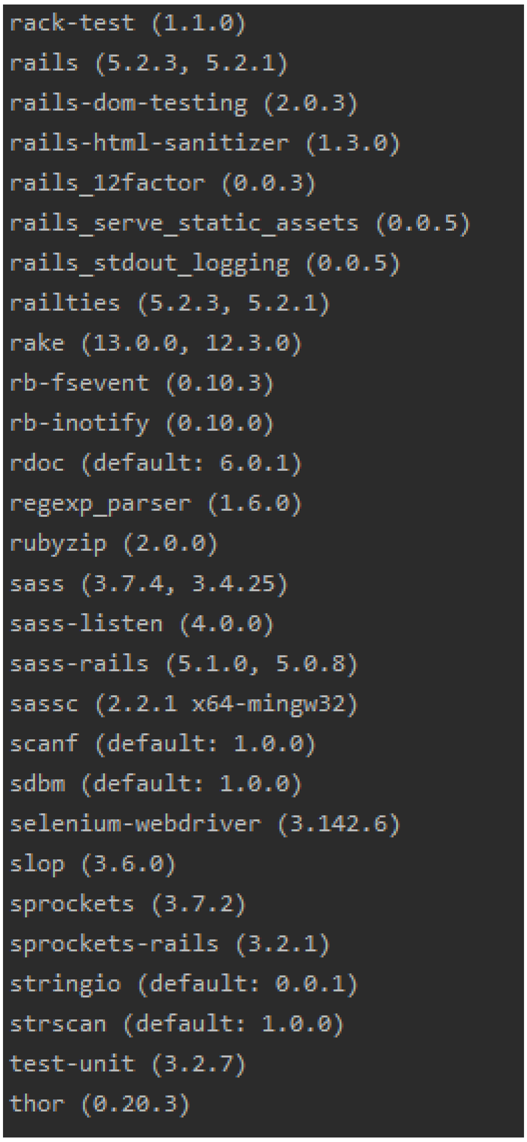
```
C:\Sites\exemplos>gem list

*** LOCAL GEMS ***

actioncable (5.2.3, 5.2.1)
actionmailer (5.2.3, 5.2.1)
actionpack (5.2.3, 5.2.1)
actionview (5.2.3, 5.2.1)
activejob (5.2.3, 5.2.1)
activemodel (5.2.3, 5.2.1)
activerecord (5.2.3, 5.2.1)
activestorage (5.2.3, 5.2.1)
activesupport (5.2.3, 5.2.1)
addressable (2.7.0)
archive-zip (0.12.0)
arel (9.0.0)
autoprefixer-rails (9.6.4)
bigdecimal (default: 1.3.4)
bindex (0.8.1)
bootsnap (1.4.5)
bootstrap-sass (3.4.1)
builder (3.2.3)
bundler (2.0.2)
byebug (11.0.1)
capybara (3.29.0)
childprocess (3.0.0)
chromedriver-helper (2.1.1)
chunky_png (1.3.11)
```



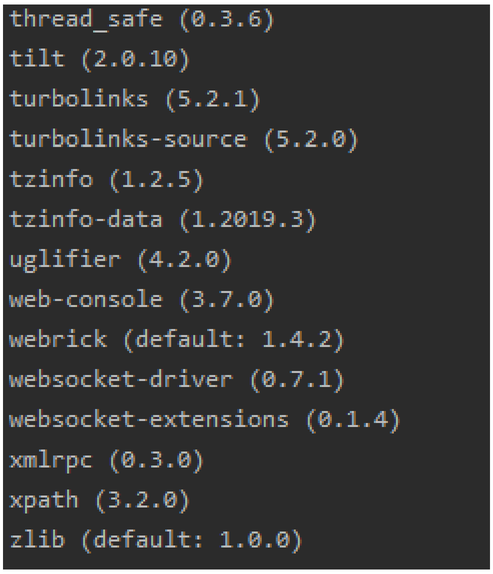
```
cmath (default: 1.0.0)
coderay (1.1.2)
coffee-rails (4.2.2)
coffee-script (2.4.1)
coffee-script-source (1.12.2)
compass (1.0.3, 0.12.2)
compass-core (1.0.3)
compass-import-once (1.0.5)
compass-rails (3.0.2, 2.0.0)
concurrent-ruby (1.1.5)
crass (1.0.4)
csv (default: 1.0.0)
date (default: 1.0.0)
dbm (default: 1.0.0)
did_you_mean (1.2.0)
duktape (2.3.0.0)
erubi (1.9.0)
etc (default: 1.0.0)
execjs (2.7.0)
fcntl (default: 1.0.0)
ffi (1.11.1 x64-mingw32)
fiddle (default: 1.0.0)
fileutils (default: 1.0.2)
fssm (0.2.10)
gdbm (default: 2.0.0)
globalid (0.4.2)
i18n (1.7.0)
io-console (default: 0.4.6)
```



```
io-like (0.3.0)
ipaddr (default: 1.2.0)
jbuilder (2.9.1)
jquery (0.0.1)
jquery-rails (4.3.5, 4.3.1)
json (default: 2.1.0)
loofah (2.3.0)
mail (2.7.1)
marcel (0.3.3)
method_source (0.9.2)
mimemagic (0.3.3)
mini_mime (1.0.2)
mini_portile2 (2.4.0)
minitest (5.12.2, 5.10.3)
msgpack (1.3.1 x64-mingw32)
multi_json (1.13.1)
mysql2 (0.5.2 x64-mingw32)
net-telnet (0.1.1)
nio4r (2.5.2)
nokogiri (1.10.4 x64-mingw32)
openssl (default: 2.1.0)
power_assert (1.1.1)
pry (0.12.2, 0.9.12.6)
pry-require_relative (0.0.2)
psych (default: 3.0.2)
public_suffix (4.0.1)
puma (3.12.1)
rack (2.0.7)
```



```
rack-test (1.1.0)
rails (5.2.3, 5.2.1)
rails-dom-testing (2.0.3)
rails-html-sanitizer (1.3.0)
rails_12factor (0.0.3)
rails_serve_static_assets (0.0.5)
rails_stdout_logging (0.0.5)
railties (5.2.3, 5.2.1)
rake (13.0.0, 12.3.0)
rb-fsevent (0.10.3)
rb-inotify (0.10.0)
rdoc (default: 6.0.1)
regexp_parser (1.6.0)
rubyzip (2.0.0)
sass (3.7.4, 3.4.25)
sass-listen (4.0.0)
sass-rails (5.1.0, 5.0.8)
sassc (2.2.1 x64-mingw32)
scanf (default: 1.0.0)
sdbm (default: 1.0.0)
selenium-webdriver (3.142.6)
slop (3.6.0)
sprockets (3.7.2)
sprockets-rails (3.2.1)
stringio (default: 0.0.1)
strscan (default: 1.0.0)
test-unit (3.2.7)
thor (0.20.3)
```



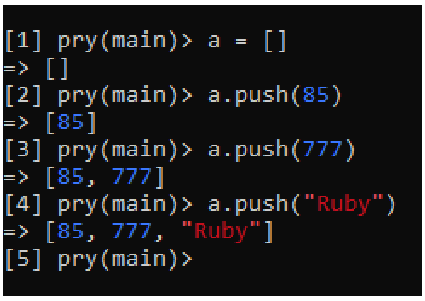
```
thread_safe (0.3.6)
tilt (2.0.10)
turbolinks (5.2.1)
turbolinks-source (5.2.0)
tzinfo (1.2.5)
tzinfo-data (1.2019.3)
uglifier (4.2.0)
web-console (3.7.0)
webrick (default: 1.4.2)
websocket-driver (0.7.1)
websocket-extensions (0.1.4)
xmlrpc (0.3.0)
xpath (3.2.0)
zlib (default: 1.0.0)
```

Aula 06 - Array, Hash, Métodos, Parâmetros e Blocos

Array

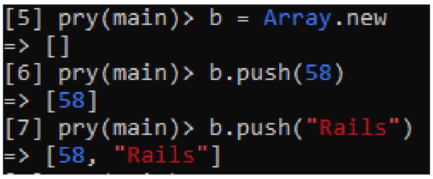
"Array é uma coleção ordenada de qualquer objeto indexado por um inteiro. Cada elemento de um array é associado e referenciado por índice."

[]



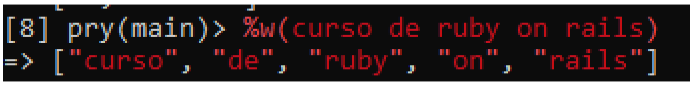
```
[1] pry(main)> a = []
=> []
[2] pry(main)> a.push(85)
=> [85]
[3] pry(main)> a.push(777)
=> [85, 777]
[4] pry(main)> a.push("Ruby")
=> [85, 777, "Ruby"]
[5] pry(main)>
```

.new



```
[5] pry(main)> b = Array.new
=> []
[6] pry(main)> b.push(58)
=> [58]
[7] pry(main)> b.push("Rails")
=> [58, "Rails"]
```

%w

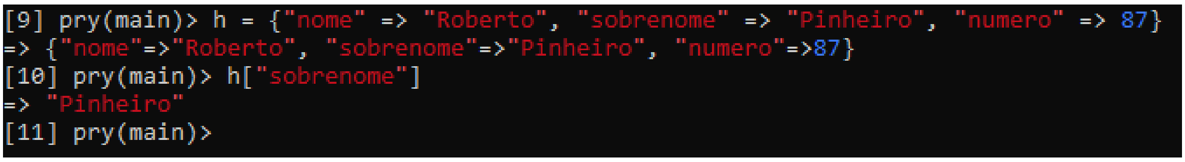


```
[8] pry(main)> %w(curso de ruby on rails)
=> ["curso", "de", "ruby", "on", "rails"]
```

Hash

"Hash é uma coleção de pares chave-valor. É similar a um array exceto por seus índices serem definidos arbitrariamente."

{:key => :value}



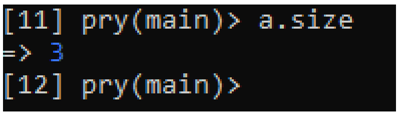
```
[9] pry(main)> h = {"nome" => "Roberto", "sobrenome" => "Pinheiro", "numero" => 87}
=> {"nome"=>"Roberto", "sobrenome"=>"Pinheiro", "numero"=>87}
[10] pry(main)> h["sobrenome"]
=> "Pinheiro"
[11] pry(main)>
```

Métodos, parâmetros e blocos

No Ruby tudo é objeto.

Métodos

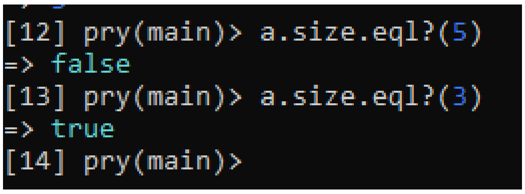
Métodos são as ações.



```
[11] pry(main)> a.size
=> 3
[12] pry(main)>
```

Parâmetros

É a passagem de um dado para um método para que seja retornado algo.

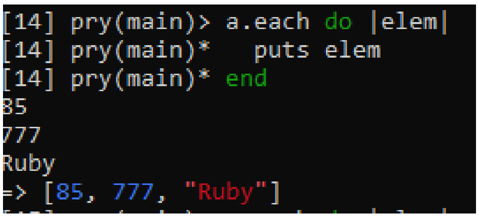


```
[12] pry(main)> a.size.eql?(5)
=> false
[13] pry(main)> a.size.eql?(3)
=> true
[14] pry(main)>
```

Obs: No Ruby, métodos podem ser encadeados.

Blocos

No bloco, entre o do |elem| e o end é inserido o código.



```
[14] pry(main)> a.each do |elem|
[14] pry(main)*   puts elem
[14] pry(main)* end
85
777
Ruby
=> [85, 777, "Ruby"]
```

Aula 07 - Classes, Módulos, Mixins e o Yield

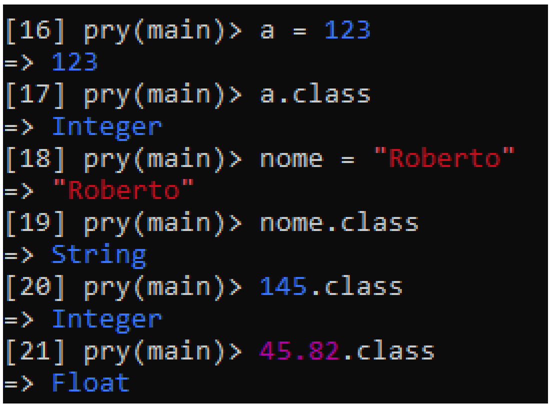
Classes

O que é uma classe?

"Na programação orientada a objetos, uma classe é uma construção que é usada para criar instâncias de si mesma."

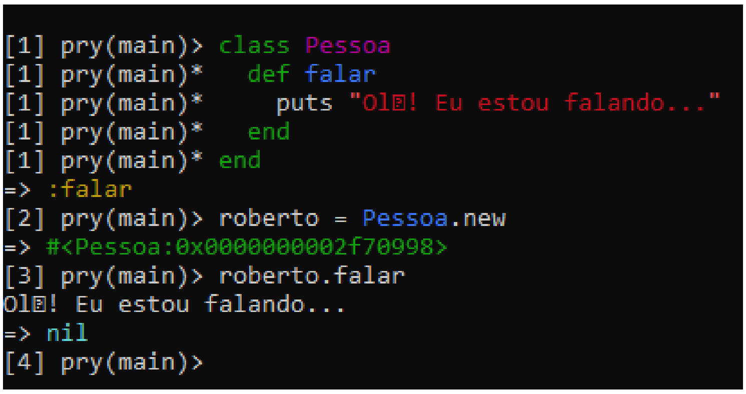
.class

No Ruby, tudo é objeto. Logo, tudo é representado por uma classe.



```
[16] pry(main)> a = 123
=> 123
[17] pry(main)> a.class
=> Integer
[18] pry(main)> nome = "Roberto"
=> "Roberto"
[19] pry(main)> nome.class
=> String
[20] pry(main)> 145.class
=> Integer
[21] pry(main)> 45.82.class
=> Float
```

Instanciando um objeto

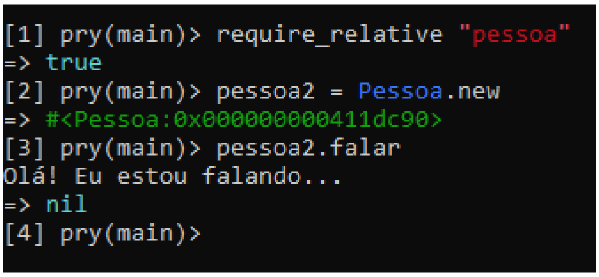


```
[1] pry(main)> class Pessoa
[1] pry(main)*   def falar
[1] pry(main)*     puts "Ol�! Eu estou falando..."
[1] pry(main)*   end
[1] pry(main)* end
=> :falar
[2] pry(main)> roberto = Pessoa.new
=> #<Pessoa:0x0000000002f70998>
[3] pry(main)> roberto.falar
Ol�! Eu estou falando...
=> nil
[4] pry(main)>
```

No irb para limpar a tela:

system 'cls'

Herança



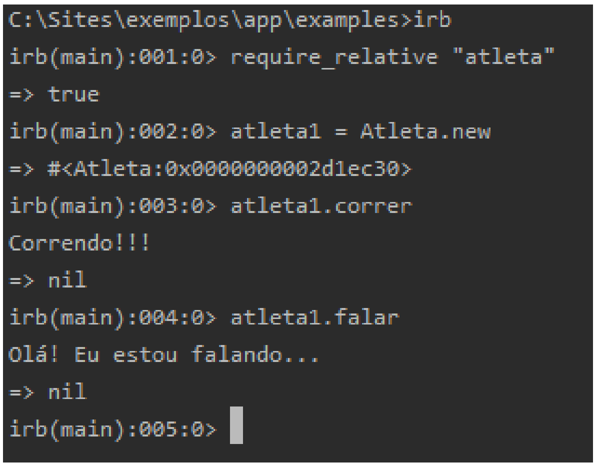
```
[1] pry(main)> require_relative "pessoa"
=> true
[2] pry(main)> pessoa2 = Pessoa.new
=> #<Pessoa:0x000000000411dc90>
[3] pry(main)> pessoa2.falar
Olá! Eu estou falando...
=> nil
[4] pry(main)>
```

app/examples/pessoa.rb

```
class Pessoa
  def falar
    puts "Olá! Eu estou falando..."
  end
end
```

app/examples/atleta.rb

```
require_relative "pessoa"

class Atleta < Pessoa
  def correr
    puts "Correndo!!!"
  end
end
```



```
C:\Sites\exemplos\app\examples>irb
irb(main):001:0> require_relative "atleta"
=> true
irb(main):002:0> atleta1 = Atleta.new
=> #<Atleta:0x0000000002d1ec30>
irb(main):003:0> atleta1.correr
Correndo!!!
=> nil
irb(main):004:0> atleta1.falar
Olá! Eu estou falando...
=> nil
irb(main):005:0>
```

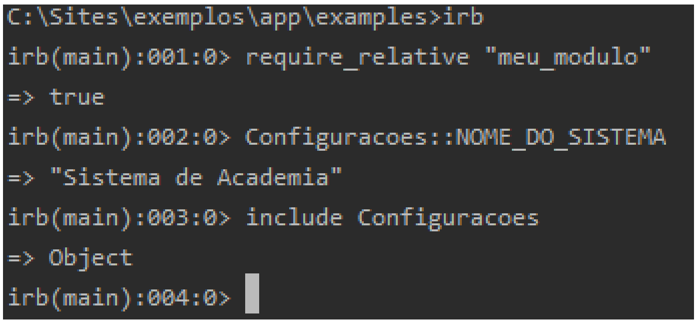
Módulo

Um módulo é uma coleção de métodos e constantes.

Constante é uma variável cujo valor não pode ser alterado.

app/examples/meu_modulo.rb

```
module Configuracoes
  NOME_DO_SISTEMA = "Sistema de Academia"
  VERSAO = "1.2.4.5"

  def calcular
    puts "O resultado final é...."
  end
end
```



```
C:\Sites\exemplos\app\examples>irb
irb(main):001:0> require_relative "meu_modulo"
=> true
irb(main):002:0> Configuracoes::NOME_DO_SISTEMA
=> "Sistema de Academia"
irb(main):003:0> include Configuracoes
=> Object
irb(main):004:0>
```

Mixin

O que é um Mixin?

"Ruby não suporta herança múltipla diretamente, mas módulos de Ruby podem ter outros usos maravilhosos. De um só golpe, eles praticamente eliminam a necessidade de herança múltipla, proporcionando um mecanismo chamado de Mixin".

app/examples/meu_modulo.rb

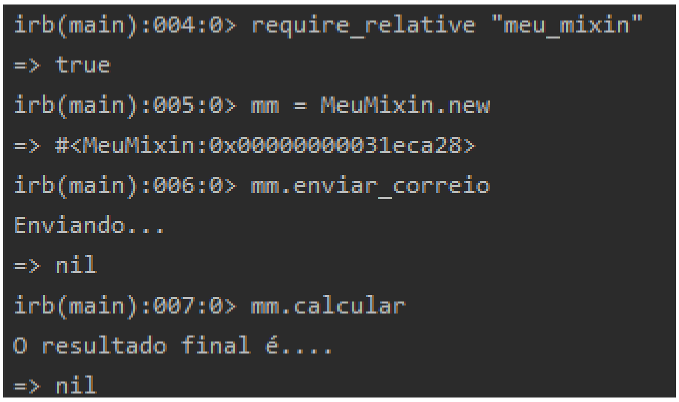
```
module Configuracoes
  NOME_DO_SISTEMA = "Sistema de Academia"
  VERSAO = "1.2.4.5"

  def calcular
    puts "O resultado final é...."
  end
end
```

app/examples/modulo_correio.rb

```
module Correio
  def enviar_correio
    puts "Enviando..."
  end
end
```

app/examples/meu_mixin.rb

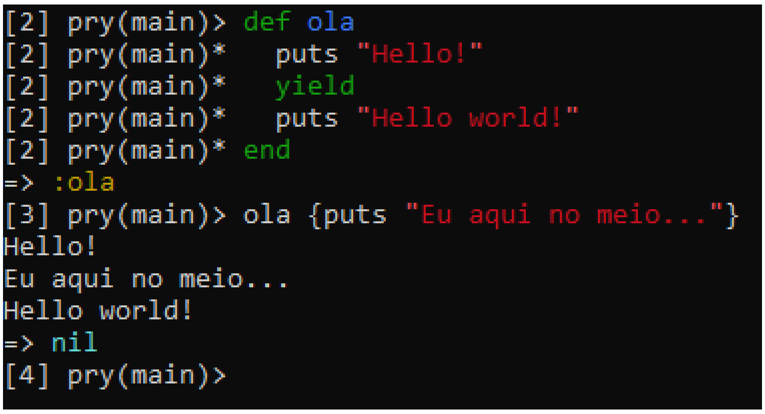
```
require_relative "meu_modulo"
require_relative "modulo_correio"

class MeuMixin
  include Configuracoes
  include Correio
end
```



```
irb(main):004:0> require_relative "meu_mixin"
=> true
irb(main):005:0> mm = MeuMixin.new
=> #<MeuMixin:0x00000000031eca28>
irb(main):006:0> mm.enviar_correio
Enviando...
=> nil
irb(main):007:0> mm.calcular
O resultado final é....
=> nil
```

Blocos e o Yield

É uma forma de passar bloco de código para dentro de uma função.



```
[2] pry(main)> def ola
[2] pry(main)*   puts "Hello!"
[2] pry(main)*   yield
[2] pry(main)*   puts "Hello world!"
[2] pry(main)* end
=> :ola
[3] pry(main)> ola {puts "Eu aqui no meio..."}
Hello!
Eu aqui no meio...
Hello world!
=> nil
[4] pry(main)>
```

Aula 08 – Banco de dados e Rake

```
rails new exemplos --database=mysql
```

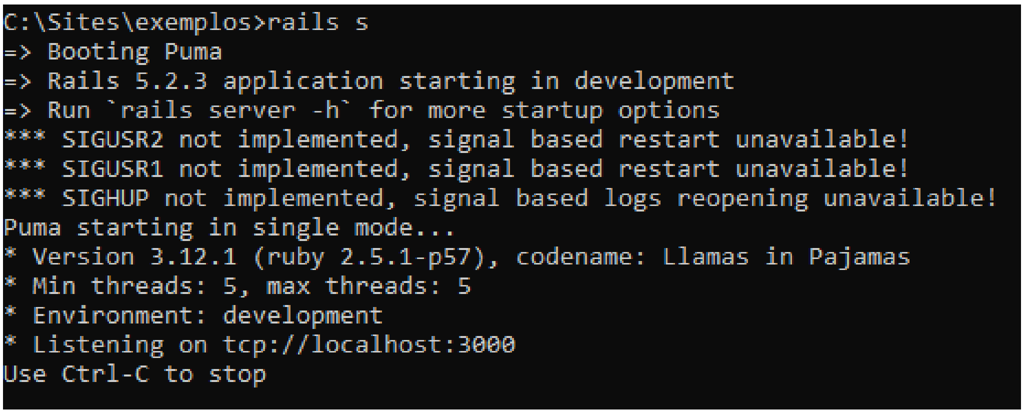
config/database.yml

```
# MySQL. Versions 5.1.10 and up are supported.
#
# Install the MySQL driver
#   gem install mysql2
#
# Ensure the MySQL gem is defined in your Gemfile
#   gem 'mysql2'
#
# And be sure to use new-style password hashing:
#   https://dev.mysql.com/doc/refman/5.7/en/password-hashing.html
#
default: &default
  adapter: mysql2
  encoding: utf8
  pool: <%= ENV.fetch("RAILS_MAX_THREADS") { 5 } %>
  username: root
  password:
  host: localhost

development:
  <<: *default
  database: exemplos_development

# Warning: The database defined as "test" will be erased and
# re-generated from your development database when you run "rake".
# Do not set this db to the same as development or production.
test:
  <<: *default
  database: exemplos_test

# As with config/secrets.yml, you never want to store sensitive information,
# like your database password, in your source code. If your source code is
# ever seen by anyone, they now have access to your database.
#
# Instead, provide the password as a unix environment variable when you boot
# the app. Read http://guides.rubyonrails.org/configuring.html#configuring-a-database
# for a full rundown on how to provide these environment variables in a
# production deployment.
#
# On Heroku and other platform providers, you may have a full connection URL
# available as an environment variable. For example:
#
```

```
#   DATABASE_URL="mysql2://myuser:mypass@localhost/somedatabase"
#
# You can use this database configuration with:
#
#   production:
#     url: <%= ENV['DATABASE_URL'] %>
#
production:
  <<: *default
  database: exemplos_production
  username: exemplos
  password: <%= ENV['EXEMPLOS_DATABASE_PASSWORD'] %>
```

O padrão do Ruby é ambiente de desenvolvimento (exemplos_development).

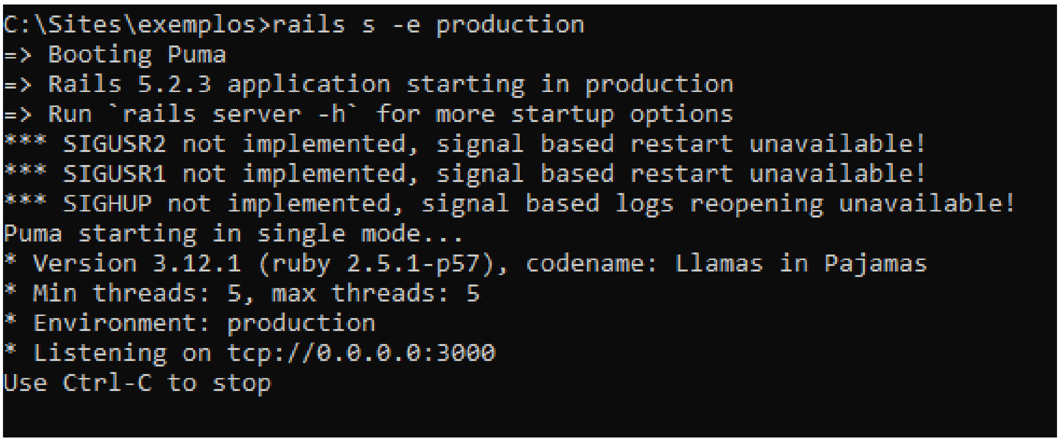
rails s



```
C:\Sites\exemplos>rails s
=> Booting Puma
=> Rails 5.2.3 application starting in development
=> Run `rails server -h` for more startup options
*** SIGUSR2 not implemented, signal based restart unavailable!
*** SIGUSR1 not implemented, signal based restart unavailable!
*** SIGHUP not implemented, signal based logs reopening unavailable!
Puma starting in single mode...
* Version 3.12.1 (ruby 2.5.1-p57), codename: Llamas in Pajamas
* Min threads: 5, max threads: 5
* Environment: development
* Listening on tcp://localhost:3000
Use Ctrl-C to stop
```

Alterando para ambiente de produção

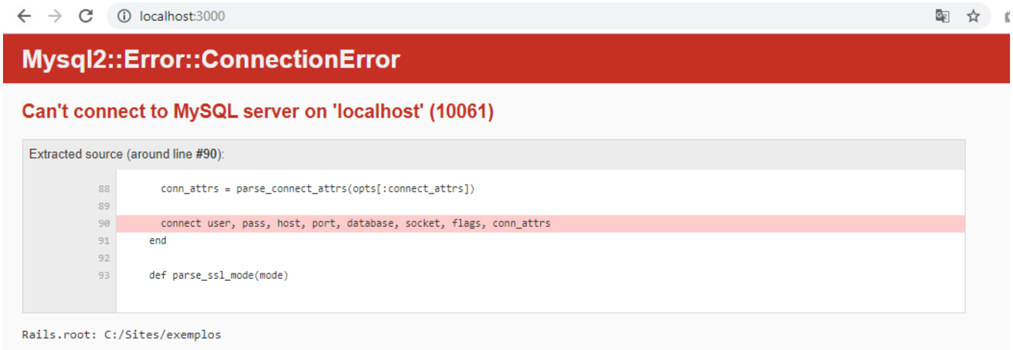
rails s -e production



```
C:\Sites\exemplos>rails s -e production
=> Booting Puma
=> Rails 5.2.3 application starting in production
=> Run `rails server -h` for more startup options
*** SIGUSR2 not implemented, signal based restart unavailable!
*** SIGUSR1 not implemented, signal based restart unavailable!
*** SIGHUP not implemented, signal based logs reopening unavailable!
Puma starting in single mode...
* Version 3.12.1 (ruby 2.5.1-p57), codename: Llamas in Pajamas
* Min threads: 5, max threads: 5
* Environment: production
* Listening on tcp://0.0.0.0:3000
Use Ctrl-C to stop
```

rails s

localhost:3000



Mysql2::Error::ConnectionError

Can't connect to MySQL server on 'localhost' (10061)

Extracted source (around line #90):

```
88        conn_attrs = parse_connect_attrs(opts[:connect_attrs])
89
90        connect user, pass, host, port, database, socket, flags, conn_attrs
91      end
92
93      def parse_ssl_mode(mode)
```

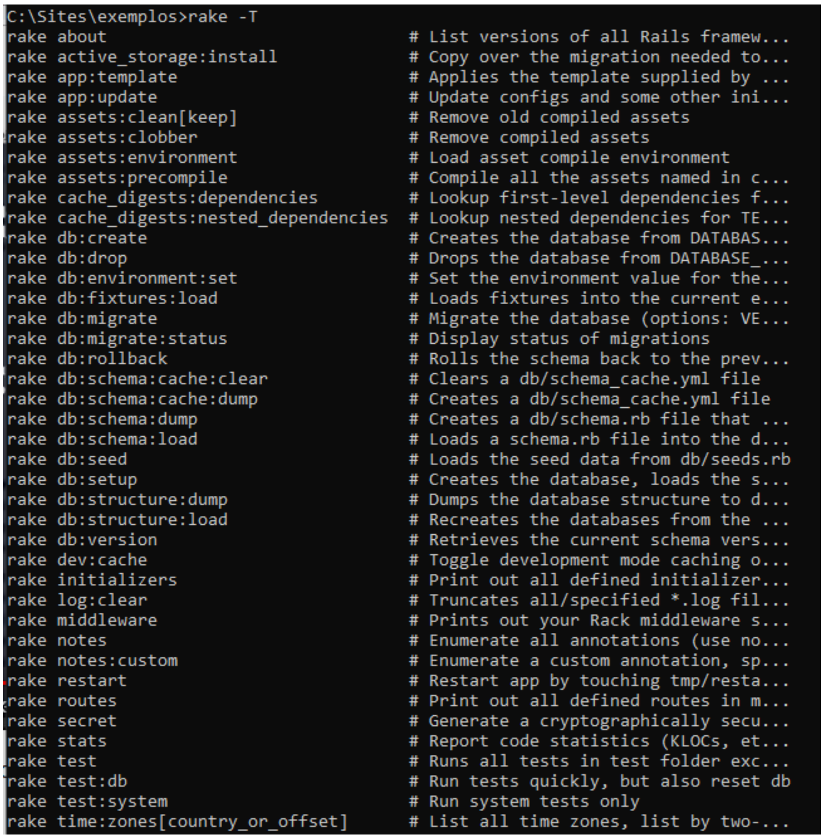
Rails.root: C:/Sites/exemplos

Esse erro ocorre porque o banco de dados ainda não foi criado.

Rake

Rake é uma ferramenta de gerenciamento de tarefas.

rake -T ou rake --tasks

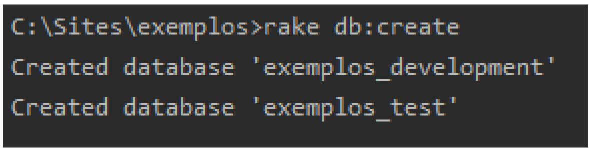


```
C:\Sites\exemplos>rake -T
rake about                              # List versions of all Rails framew...
rake active_storage:install             # Copy over the migration needed to...
rake app:template                       # Applies the template supplied by ...
rake app:update                         # Update configs and some other ini...
rake assets:clean[keep]                 # Remove old compiled assets
rake assets:clobber                     # Remove compiled assets
rake assets:environment                 # Load asset compile environment
rake assets:precompile                  # Compile all the assets named in c...
rake cache_digests:dependencies         # Lookup first-level dependencies f...
rake cache_digests:nested_dependencies  # Lookup nested dependencies for TE...
rake db:create                          # Creates the database from DATABAS...
rake db:drop                            # Drops the database from DATABASE_...
rake db:environment:set                 # Set the environment value for the...
rake db:fixtures:load                   # Loads fixtures into the current e...
rake db:migrate                         # Migrate the database (options: VE...
rake db:migrate:status                  # Display status of migrations
rake db:rollback                        # Rolls the schema back to the prev...
rake db:schema:cache:clear              # Clears a db/schema_cache.yml file
rake db:schema:cache:dump               # Creates a db/schema_cache.yml file
rake db:schema:dump                     # Creates a db/schema.rb file that ...
rake db:schema:load                     # Loads a schema.rb file into the d...
rake db:seed                            # Loads the seed data from db/seeds.rb
rake db:setup                           # Creates the database, loads the s...
rake db:structure:dump                  # Dumps the database structure to d...
rake db:structure:load                  # Recreates the databases from the ...
rake db:version                         # Retrieves the current schema vers...
rake dev:cache                          # Toggle development mode caching o...
rake initializers                       # Print out all defined initializer...
rake log:clear                          # Truncates all/specified *.log fil...
rake middleware                         # Prints out your Rack middleware s...
rake notes                              # Enumerate all annotations (use no...
rake notes:custom                       # Enumerate a custom annotation, sp...
rake restart                            # Restart app by touching tmp/resta...
rake routes                             # Print out all defined routes in m...
rake secret                             # Generate a cryptographically secu...
rake stats                              # Report code statistics (KLOCs, et...
rake test                               # Runs all tests in test folder exc...
rake test:db                            # Run tests quickly, but also reset db
rake test:system                        # Run system tests only
rake time:zones[country_or_offset]      # List all time zones, list by two-...
```

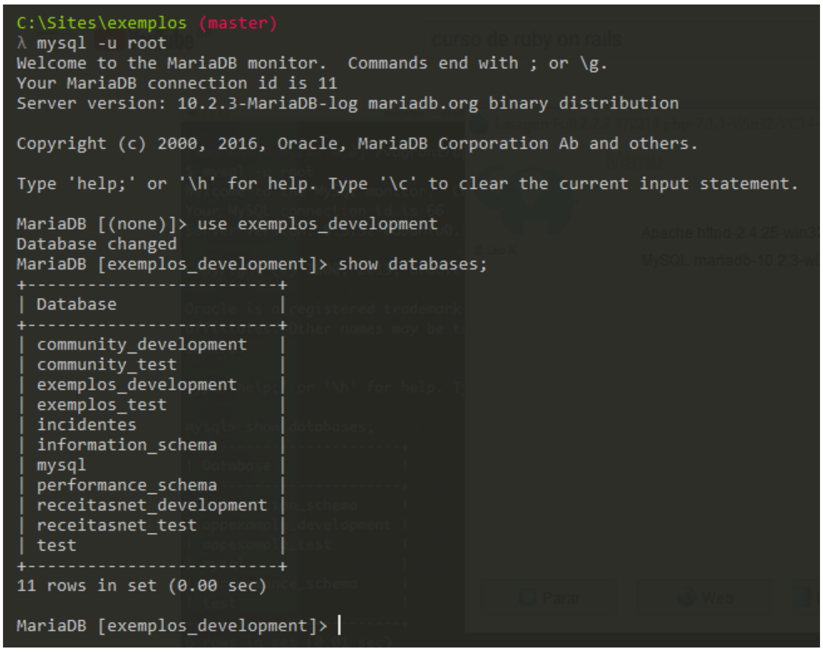
Criando o banco de dados

Com o MySQL rodando, entre no terminal, com o comando:

rake db:create



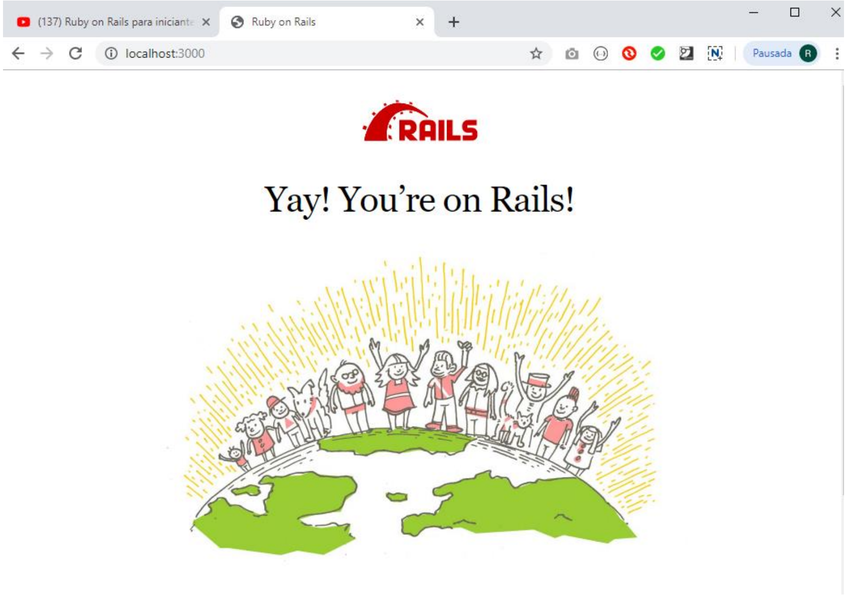
```
C:\Sites\exemplos>rake db:create
Created database 'exemplos_development'
Created database 'exemplos_test'
```



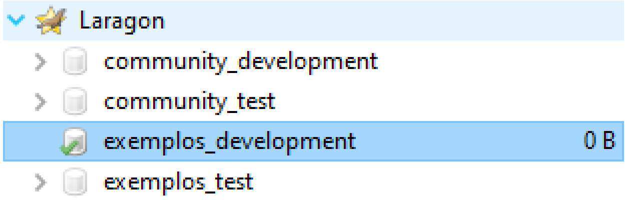
```
C:\Sites\exemplos (master)
λ mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 11
Server version: 10.2.3-MariaDB-log mariadb.org binary distribution

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use exemplos_development
Database changed
MariaDB [exemplos_development]> show databases;
+-------------------------+
| Database                |
+-------------------------+
| community_development   |
| community_test          |
| exemplos_development    |
| exemplos_test           |
| incidentes              |
| information_schema      |
| mysql                   |
| performance_schema      |
| receitasnet_development |
| receitasnet_test        |
| test                    |
+-------------------------+
11 rows in set (0.00 sec)

MariaDB [exemplos_development]> |
```

No navegador:

localhost:3000



Yay! You're on Rails!



Laragon

community_development

community_test

exemplos_development 0 B

exemplos_test

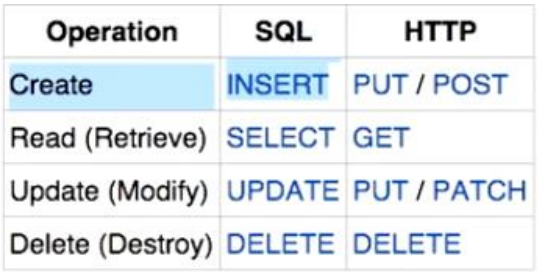
Aula 09 – CoC e criando o primeiro scaffold

CoC – Convention on Configuration

Convenção sobre configuração (CoC) é utilizado para diminuir o número de decisões que o desenvolvedor necessita fazer, ganhando simplicidade, mas não necessariamente perdendo flexibilidade.

CRUD

São as quatro operações básicas.



Operation	SQL	HTTP
Create	INSERT	PUT / POST
Read (Retrieve)	SELECT	GET
Update (Modify)	UPDATE	PUT / PATCH
Delete (Destroy)	DELETE	DELETE

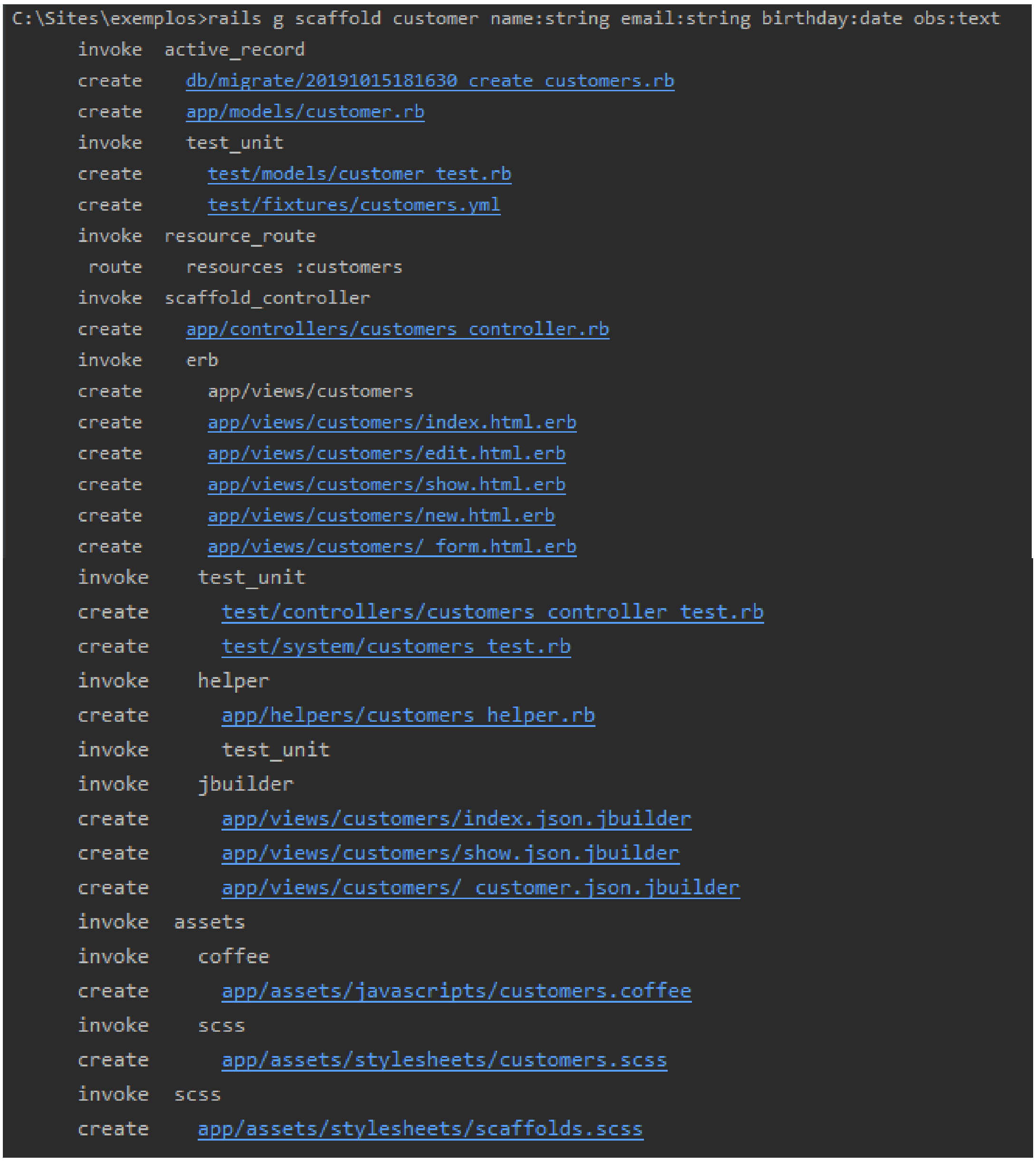
Scaffold

Ao se gerar um scaffold cria-se um CRUD.

Gerando um scaffold no Rails

Na linha de comando, é especificado o nome do modelo (no caso, customer).

rails g scaffold customer name:string email:string birthday:date obs:text



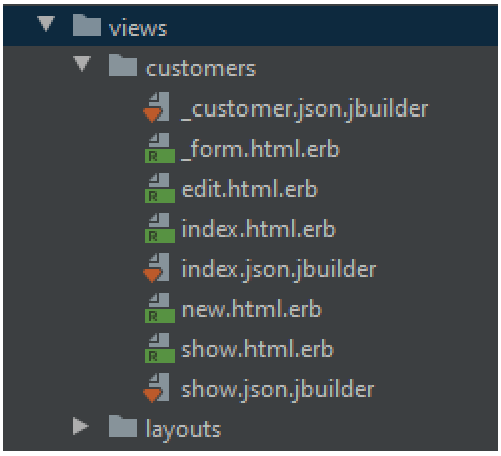
```
C:\Sites\exemplos>rails g scaffold customer name:string email:string birthday:date obs:text
      invoke  active_record
      create    db/migrate/20191015181630_create_customers.rb
      create    app/models/customer.rb
      invoke    test_unit
      create      test/models/customer_test.rb
      create      test/fixtures/customers.yml
      invoke  resource_route
       route    resources :customers
      invoke  scaffold_controller
      create    app/controllers/customers_controller.rb
      invoke    erb
      create      app/views/customers
      create      app/views/customers/index.html.erb
      create      app/views/customers/edit.html.erb
      create      app/views/customers/show.html.erb
      create      app/views/customers/new.html.erb
      create      app/views/customers/_form.html.erb
      invoke    test_unit
      create      test/controllers/customers_controller_test.rb
      create      test/system/customers_test.rb
      invoke    helper
      create      app/helpers/customers_helper.rb
      invoke      test_unit
      invoke    jbuilder
      create      app/views/customers/index.json.jbuilder
      create      app/views/customers/show.json.jbuilder
      create      app/views/customers/_customer.json.jbuilder
      invoke  assets
      invoke    coffee
      create      app/assets/javascripts/customers.coffee
      invoke    scss
      create      app/assets/stylesheets/customers.scss
      invoke  scss
      create    app/assets/stylesheets/scaffolds.scss
```

app/controllers/customers_controller.rb

```
class CustomersController < ApplicationController
  before_action :set_customer, only: [:show, :edit, :update, :destroy]

  # GET /customers
  # GET /customers.json
  def index
    @customers = Customer.all
  end

  # GET /customers/1
  # GET /customers/1.json
  def show
  end

  # GET /customers/new
  def new
    @customer = Customer.new
  end

  # GET /customers/1/edit
  def edit
  end

  # POST /customers
  # POST /customers.json
  def create
    @customer = Customer.new(customer_params)

    respond_to do |format|
      if @customer.save
        format.html { redirect_to @customer, notice: 'Customer was successfully created.' }
        format.json { render :show, status: :created, location: @customer }
      else
        format.html { render :new }
        format.json { render json: @customer.errors, status: :unprocessable_entity }
      end
    end
  end

  # PATCH/PUT /customers/1
  # PATCH/PUT /customers/1.json
  def update
    respond_to do |format|
      if @customer.update(customer_params)
        format.html { redirect_to @customer, notice: 'Customer was successfully updated.' }
        format.json { render :show, status: :ok, location: @customer }
      else
        format.html { render :edit }
        format.json { render json: @customer.errors, status: :unprocessable_entity }
      end
    end
  end

  # DELETE /customers/1
  # DELETE /customers/1.json
  def destroy
    @customer.destroy
    respond_to do |format|
      format.html { redirect_to customers_url, notice: 'Customer was successfully destroyed.' }
      format.json { head :no_content }
    end
  end

  private
    # Use callbacks to share common setup or constraints between actions.
    def set_customer
      @customer = Customer.find(params[:id])
    end

    # Never trust parameters from the scary internet, only allow the white list through.
    def customer_params
      params.require(:customer).permit(:name, :email, :birthday, :obs)
    end
end
```

db/migrate/20191016192643_create_customers.rb

```
class CreateCustomers < ActiveRecord::Migration[5.2]
  def change
    create_table :customers do |t|
      t.string :name
      t.string :email
      t.date :birthday
      t.text :obs

      t.timestamps
    end
  end
end
```

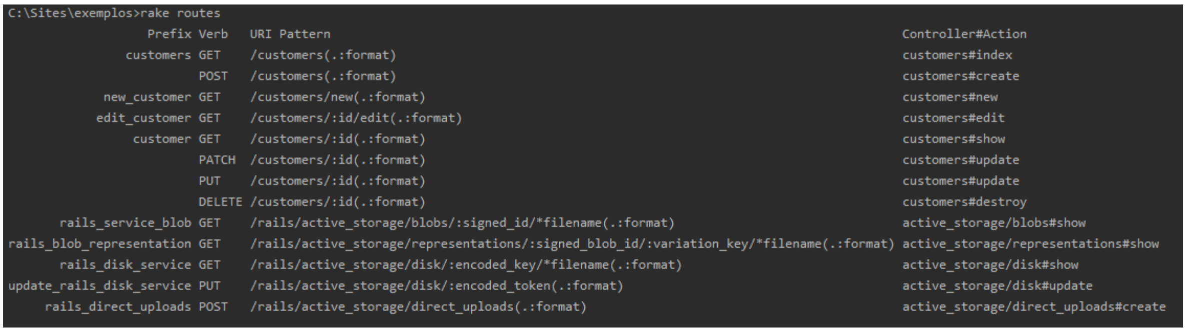


```
▼ views
    ▼ customers
        _customer.json.jbuilder
        _form.html.erb
        edit.html.erb
        index.html.erb
        index.json.jbuilder
        new.html.erb
        show.html.erb
        show.json.jbuilder
    ▶ layouts
```

config/routes.rb

```
Rails.application.routes.draw do
  resources :customers
  # For details on the DSL available within this file, see
http://guides.rubyonrails.org/routing.html
end
```

rake routes

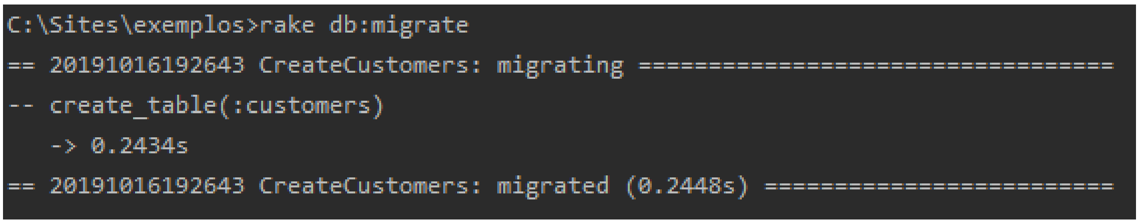


```
C:\Sites\exemplos>rake routes
                   Prefix Verb   URI Pattern                                                                              Controller#Action
                customers GET    /customers(.:format)                                                                     customers#index
                          POST   /customers(.:format)                                                                     customers#create
             new_customer GET    /customers/new(.:format)                                                                 customers#new
            edit_customer GET    /customers/:id/edit(.:format)                                                            customers#edit
                 customer GET    /customers/:id(.:format)                                                                 customers#show
                          PATCH  /customers/:id(.:format)                                                                 customers#update
                          PUT    /customers/:id(.:format)                                                                 customers#update
                          DELETE /customers/:id(.:format)                                                                 customers#destroy
       rails_service_blob GET    /rails/active_storage/blobs/:signed_id/*filename(.:format)                               active_storage/blobs#show
rails_blob_representation GET    /rails/active_storage/representations/:signed_blob_id/:variation_key/*filename(.:format) active_storage/representations#show
       rails_disk_service GET    /rails/active_storage/disk/:encoded_key/*filename(.:format)                              active_storage/disk#show
update_rails_disk_service PUT    /rails/active_storage/disk/:encoded_token(.:format)                                      active_storage/disk#update
     rails_direct_uploads POST   /rails/active_storage/direct_uploads(.:format)                                           active_storage/direct_uploads#create
```

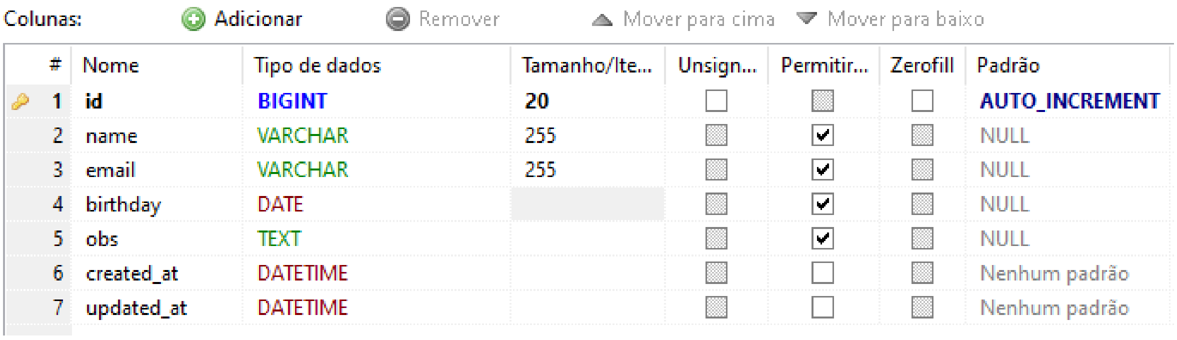
Aula 10 - Migrations, analisando e usando o Scaffold

Criando a tabela customers

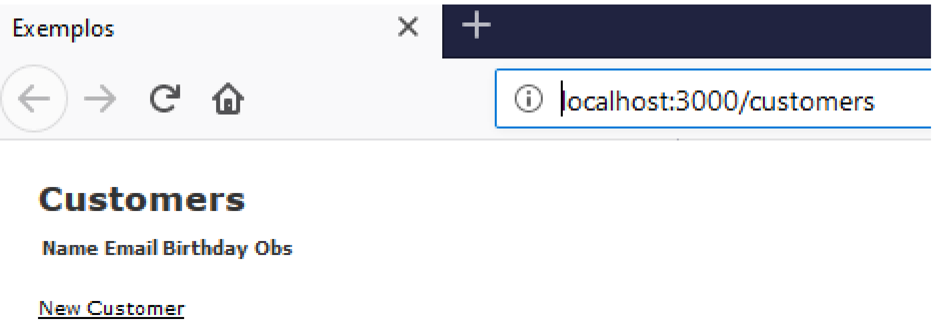
rake db:migrate



```
C:\Sites\exemplos>rake db:migrate
== 20191016192643 CreateCustomers: migrating ==================================
-- create_table(:customers)
   -> 0.2434s
== 20191016192643 CreateCustomers: migrated (0.2448s) =========================
```



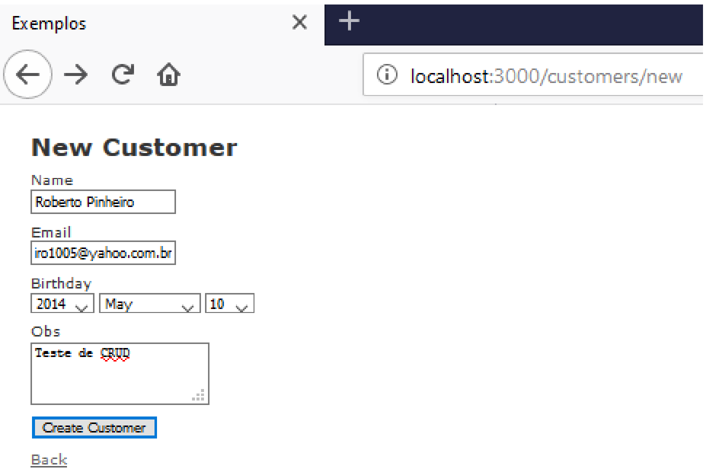
#	Nome	Tipo de dados	Tamanho/Ite...	Unsign...	Permitir...	Zerofill	Padrão
1	id	BIGINT	20	□	▨	□	AUTO_INCREMENT
2	name	VARCHAR	255	▨	☑	▨	NULL
3	email	VARCHAR	255	▨	☑	▨	NULL
4	birthday	DATE		▨	☑	▨	NULL
5	obs	TEXT		▨	☑	▨	NULL
6	created_at	DATETIME		▨	□	▨	Nenhum padrão
7	updated_at	DATETIME		▨	□	▨	Nenhum padrão



Customers

Name Email Birthday Obs

New Customer



New Customer

Name
Roberto Pinheiro

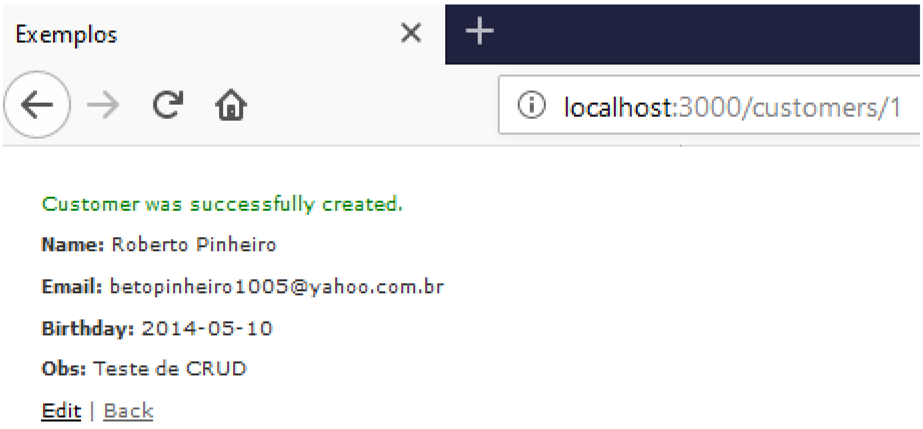
Email
iro1005@yahoo.com.br

Birthday
2014 May 10

Obs
Teste de CRUD

Create Customer

Back



Customer was successfully created.

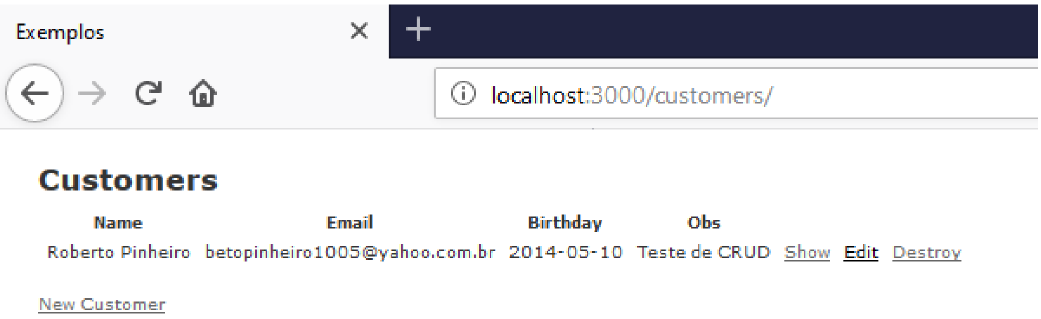
Name: Roberto Pinheiro

Email: betopinheiro1005@yahoo.com.br

Birthday: 2014-05-10

Obs: Teste de CRUD

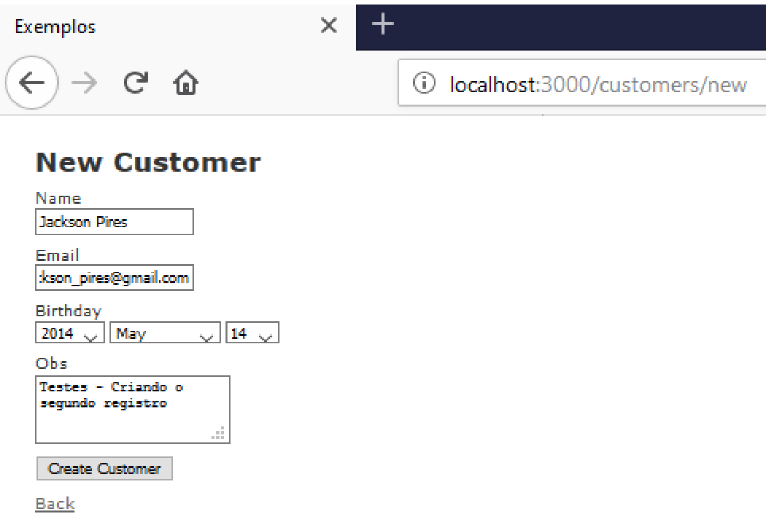
Edit | Back



Customers

Name	Email	Birthday	Obs			
Roberto Pinheiro	betopinheiro1005@yahoo.com.br	2014-05-10	Teste de CRUD	Show	Edit	Destroy

New Customer



New Customer

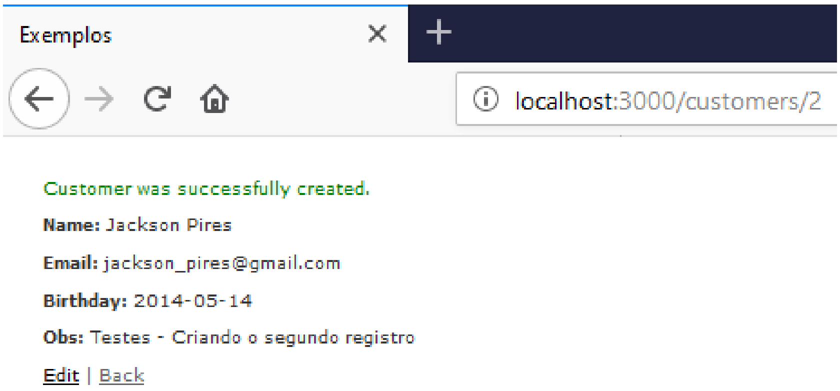
Name
Jackson Pires

Email
:kson_pires@gmail.com

Birthday
2014 | May | 14

Obs
Testes - Criando o segundo registro

Back



Customer was successfully created.

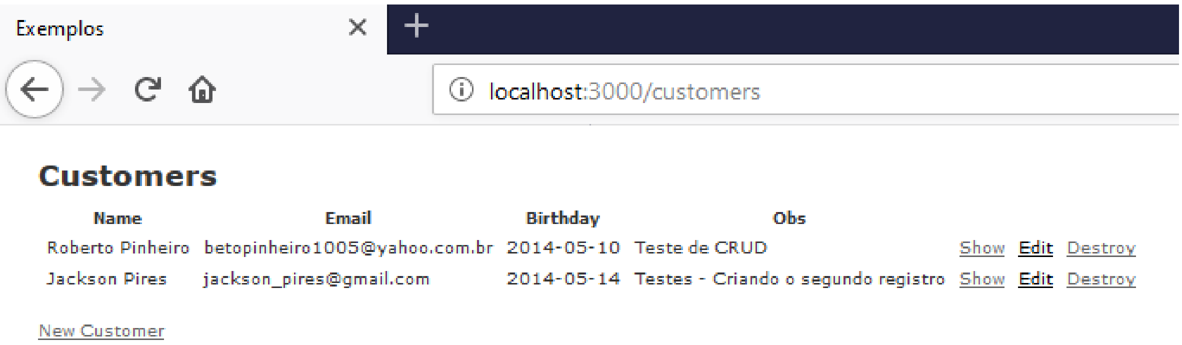
Name: Jackson Pires

Email: jackson_pires@gmail.com

Birthday: 2014-05-14

Obs: Testes - Criando o segundo registro

Edit | Back



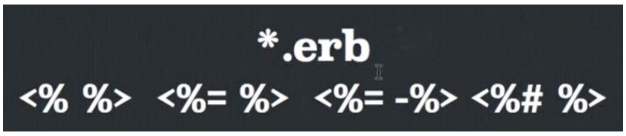
Customers

Name	Email	Birthday	Obs			
Roberto Pinheiro	betopinheiro1005@yahoo.com.br	2014-05-10	Teste de CRUD	Show	Edit	Destroy
Jackson Pires	jackson_pires@gmail.com	2014-05-14	Testes - Criando o segundo registro	Show	Edit	Destroy

New Customer

Aula 11 - Views, Embedded Ruby e Textos

erb (Embedded Ruby)



*.erb

<% %> <%= %> <%= -%> <%# %>

<% %>

Para código Ruby. Não tem saída para o HTML.

<%= %>

Para imprimir textos de variáveis Ruby. Modifica o HTML.

<%= -%>

Imprime texto eliminando espaço em branco do lado direito (tira a quebra de linha).

<%# %>

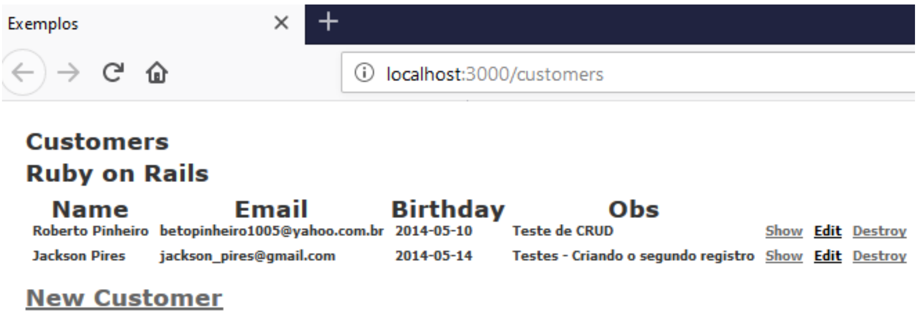
Comentário

Interpolação de variáveis de texto

#{}

app/views/customers/index.html.erb

```
<p id="notice"><%= notice %></p>

<h1>Customers</h1>

<% a = "Ruby" %>
<% b = "Rails" %>

<h1><%= "#{a} on #{b}" %></h1><h1>

<table>
  <thead>
    <tr>
      <th>Name</th>
      <th>Email</th>
      <th>Birthday</th>
      <th>Obs</th>
      <th colspan="3"></th>
    </tr>
  </thead>

  <tbody>
    <% @customers.each do |customer| %>
      <tr>
        <td><%= customer.name %></td>
        <td><%= customer.email %></td>
        <td><%= customer.birthday %></td>
        <td><%= customer.obs %></td>
        <td><%= link_to 'Show', customer %></td>
        <td><%= link_to 'Edit', edit_customer_path(customer) %></td>
        <td><%= link_to 'Destroy', customer, method: :delete, data: { confirm: 'Are you sure?' } %></td>
      </tr>
    <% end %>
  </tbody>
</table>

<br>

<%= link_to 'New Customer', new_customer_path %>
```



Customers
Ruby on Rails

Name	Email	Birthday	Obs			
Roberto Pinheiro	betopinheiro1005@yahoo.com.br	2014-05-10	Teste de CRUD	Show	Edit	Destroy
Jackson Pires	jackson_pires@gmail.com	2014-05-14	Testes - Criando o segundo registro	Show	Edit	Destroy

New Customer

Aula 12 - Active Record, Rails Console e Variáveis de Instância

Models

Model < Active Record::Base

app/models/customizer.rb

```
class Customer < ApplicationRecord
end
```

app/models/application_record.rb

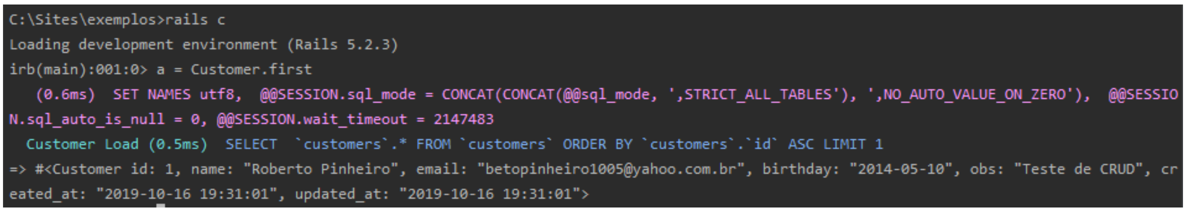
```
class ApplicationRecord < ActiveRecord::Base
  self.abstract_class = true
end
```

Console do Ruby

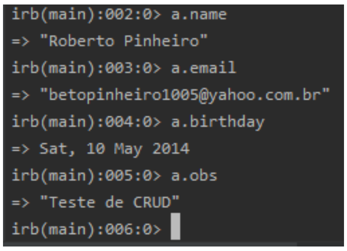
Carrega todas as classes existentes no projeto.

rails console ou rails c

a = Customer.first

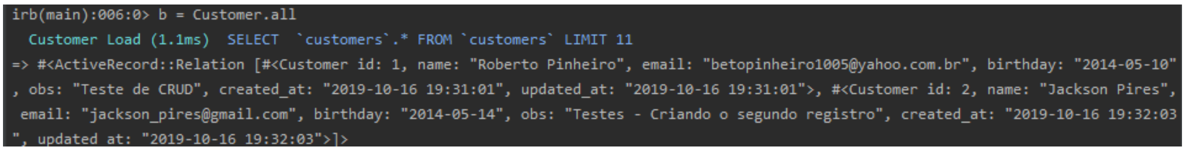


```
C:\Sites\exemplos>rails c
Loading development environment (Rails 5.2.3)
irb(main):001:0> a = Customer.first
   (0.6ms)  SET NAMES utf8,  @@SESSION.sql_mode = CONCAT(CONCAT(@@sql_mode, ',STRICT_ALL_TABLES'), ',NO_AUTO_VALUE_ON_ZERO'),  @@SESSIO
N.sql_auto_is_null = 0, @@SESSION.wait_timeout = 2147483
  Customer Load (0.5ms)  SELECT  `customers`.* FROM `customers` ORDER BY `customers`.`id` ASC LIMIT 1
=> #<Customer id: 1, name: "Roberto Pinheiro", email: "betopinheiro1005@yahoo.com.br", birthday: "2014-05-10", obs: "Teste de CRUD", cr
eated_at: "2019-10-16 19:31:01", updated_at: "2019-10-16 19:31:01">
```



```
irb(main):002:0> a.name
=> "Roberto Pinheiro"
irb(main):003:0> a.email
=> "betopinheiro1005@yahoo.com.br"
irb(main):004:0> a.birthday
=> Sat, 10 May 2014
irb(main):005:0> a.obs
=> "Teste de CRUD"
irb(main):006:0>
```

b = Customer.all



```
irb(main):006:0> b = Customer.all
  Customer Load (1.1ms)  SELECT  `customers`.* FROM `customers` LIMIT 11
=> #<ActiveRecord::Relation [#<Customer id: 1, name: "Roberto Pinheiro", email: "betopinheiro1005@yahoo.com.br", birthday: "2014-05-10"
, obs: "Teste de CRUD", created_at: "2019-10-16 19:31:01", updated_at: "2019-10-16 19:31:01">, #<Customer id: 2, name: "Jackson Pires",
 email: "jackson_pires@gmail.com", birthday: "2014-05-14", obs: "Testes - Criando o segundo registro", created_at: "2019-10-16 19:32:03
", updated_at: "2019-10-16 19:32:03">]>
```

Variáveis de instância

Variáveis de instância possuem o @ precedendo o nome da variável.

Exemplo:

No controller:

```
  def index
    @customers = Customer.all
  end
```

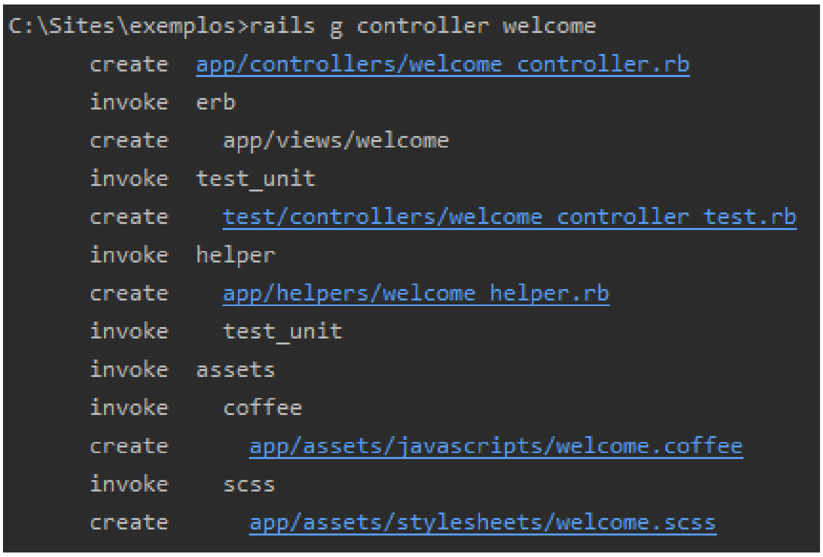
Na view "app/views/customers/index.html.erb":

```
    <% @customers.each do |customer| %>
      <tr>
        <td><%= customer.name %></td>
        <td><%= customer.email %></td>
        <td><%= customer.birthday %></td>
        <td><%= customer.obs %></td>
        <td><%= link_to 'Show', customer %></td>
        <td><%= link_to 'Edit', edit_customer_path(customer) %></td>
        <td><%= link_to 'Destroy', customer, method: :delete, data: { confirm: 'Are you sure?' } %></td>
      </tr>
    <% end %>
```

Aula 13 - Rotas, Criando um Controller e Usando Helpers

Criando um controller

rails g controller welcome



```
C:\Sites\exemplos>rails g controller welcome
      create  app/controllers/welcome_controller.rb
      invoke  erb
      create    app/views/welcome
      invoke  test_unit
      create    test/controllers/welcome_controller_test.rb
      invoke  helper
      create    app/helpers/welcome_helper.rb
      invoke    test_unit
      invoke  assets
      invoke    coffee
      create      app/assets/javascripts/welcome.coffee
      invoke    scss
      create      app/assets/stylesheets/welcome.scss
```

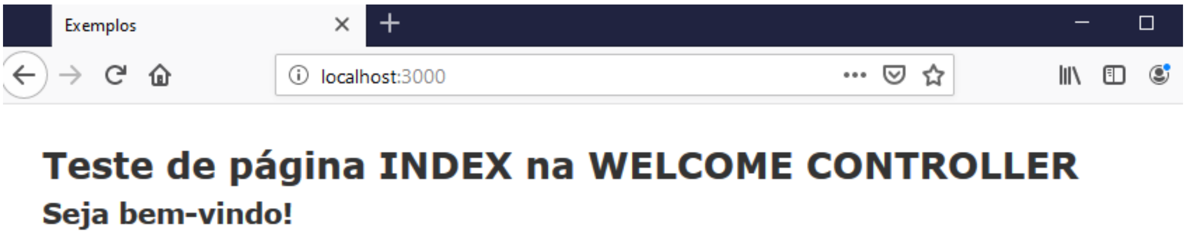
app/controllers/welcome_controller.rb

```
class WelcomeController < ApplicationController
  def index
    @message = "Seja bem-vindo!"
  end
end
```

Tornando uma rota específica como inicial

config/routes.rb

```
Rails.application.routes.draw do
  root 'welcome#index'
  get 'welcome' => 'welcome#index'
  resources :customers
end
```



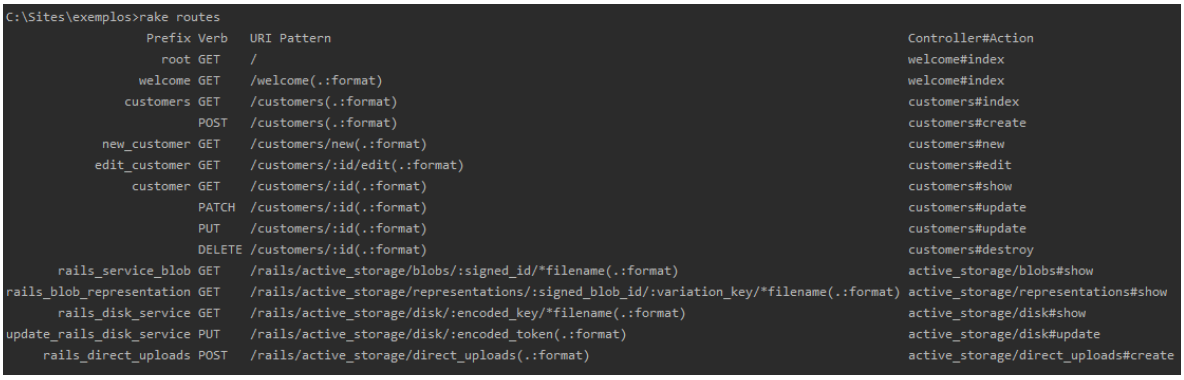
Teste de página INDEX na WELCOME CONTROLLER

Seja bem-vindo!

Helpers

link_to

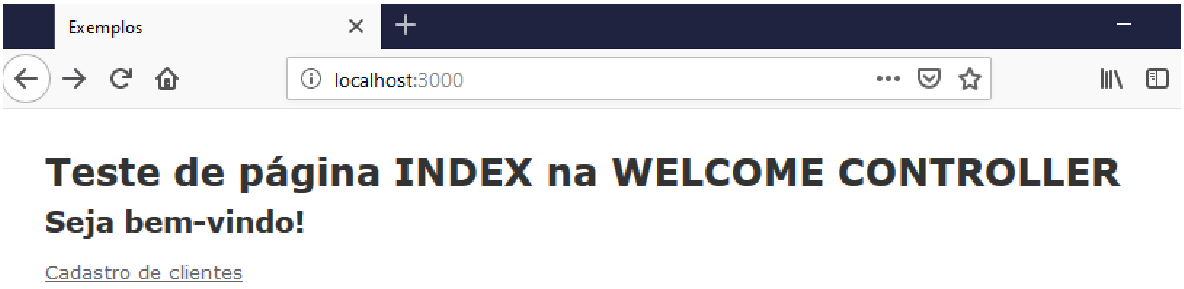
rake routes



```
C:\Sites\exemplos>rake routes
                   Prefix Verb   URI Pattern                                                                              Controller#Action
                     root GET    /                                                                                        welcome#index
                  welcome GET    /welcome(.:format)                                                                       welcome#index
                customers GET    /customers(.:format)                                                                     customers#index
                          POST   /customers(.:format)                                                                     customers#create
             new_customer GET    /customers/new(.:format)                                                                 customers#new
            edit_customer GET    /customers/:id/edit(.:format)                                                            customers#edit
                 customer GET    /customers/:id(.:format)                                                                 customers#show
                          PATCH  /customers/:id(.:format)                                                                 customers#update
                          PUT    /customers/:id(.:format)                                                                 customers#update
                          DELETE /customers/:id(.:format)                                                                 customers#destroy
       rails_service_blob GET    /rails/active_storage/blobs/:signed_id/*filename(.:format)                               active_storage/blobs#show
rails_blob_representation GET    /rails/active_storage/representations/:signed_blob_id/:variation_key/*filename(.:format) active_storage/representations#show
       rails_disk_service GET    /rails/active_storage/disk/:encoded_key/*filename(.:format)                              active_storage/disk#show
update_rails_disk_service PUT    /rails/active_storage/disk/:encoded_token(.:format)                                      active_storage/disk#update
     rails_direct_uploads POST   /rails/active_storage/direct_uploads(.:format)                                           active_storage/direct_uploads#create
```

app/views/customers/index.html.erb

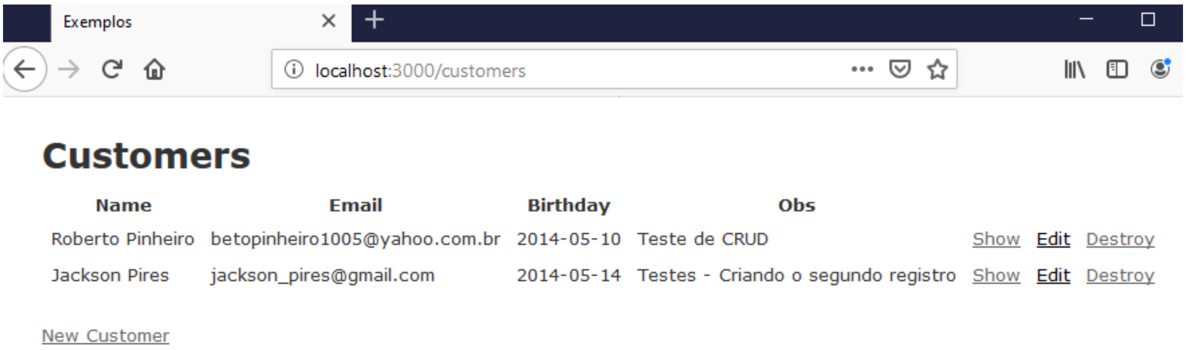
```
<h1>Teste de página INDEX na WELCOME CONTROLLER</h1>
<h2><%= @message %></h2>
<%= link_to "Cadastro de clientes", customers_path %>
```



Teste de página INDEX na WELCOME CONTROLLER

Seja bem-vindo!

Cadastro de clientes



Customers

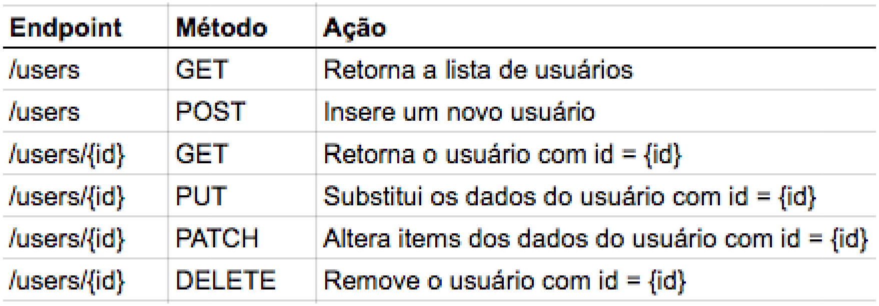
Name	Email	Birthday	Obs			
Roberto Pinheiro	betopinheiro1005@yahoo.com.br	2014-05-10	Teste de CRUD	Show	Edit	Destroy
Jackson Pires	jackson_pires@gmail.com	2014-05-14	Testes - Criando o segundo registro	Show	Edit	Destroy

New Customer

Aula 14 – Rotas REST / Restfull

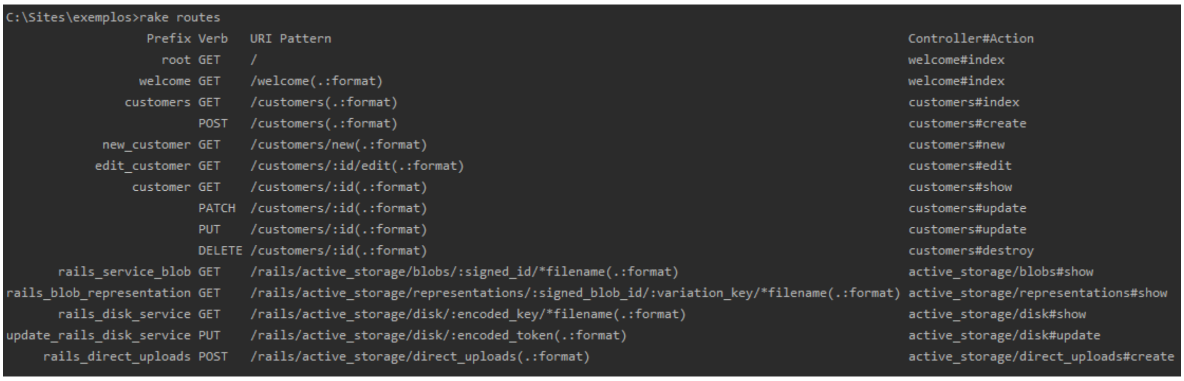
O termo REST se referia, originalmente, a um conjunto de princípios de arquitetura (descritos mais abaixo), na atualidade se usa no sentido mais amplo para descrever qualquer interface web simples que utiliza XML e HTTP (ou YAML, JSON, ou texto puro), sem as abstrações adicionais dos protocolos baseados em padrões de trocas de mensagem como o protocolo de serviços web SOAP.

Os sistemas que seguem os princípios REST são frequentemente chamados de RESTfull.



Endpoint	Método	Ação
/users	GET	Retorna a lista de usuários
/users	POST	Insere um novo usuário
/users/{id}	GET	Retorna o usuário com id = {id}
/users/{id}	PUT	Substitui os dados do usuário com id = {id}
/users/{id}	PATCH	Altera items dos dados do usuário com id = {id}
/users/{id}	DELETE	Remove o usuário com id = {id}

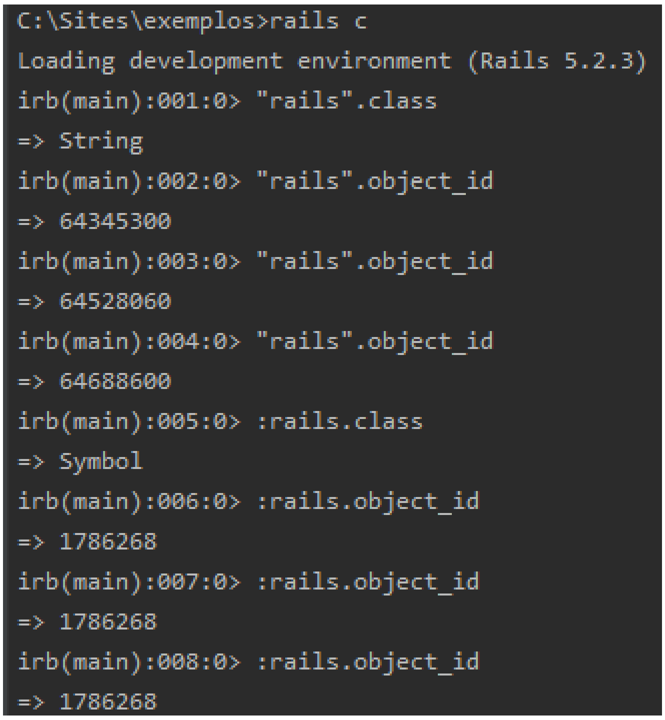
rake routes



```
C:\Sites\exemplos>rake routes
                   Prefix Verb   URI Pattern                                                                              Controller#Action
                     root GET    /                                                                                        welcome#index
                  welcome GET    /welcome(.:format)                                                                       welcome#index
                customers GET    /customers(.:format)                                                                     customers#index
                          POST   /customers(.:format)                                                                     customers#create
             new_customer GET    /customers/new(.:format)                                                                 customers#new
            edit_customer GET    /customers/:id/edit(.:format)                                                            customers#edit
                 customer GET    /customers/:id(.:format)                                                                 customers#show
                          PATCH  /customers/:id(.:format)                                                                 customers#update
                          PUT    /customers/:id(.:format)                                                                 customers#update
                          DELETE /customers/:id(.:format)                                                                 customers#destroy
       rails_service_blob GET    /rails/active_storage/blobs/:signed_id/*filename(.:format)                               active_storage/blobs#show
rails_blob_representation GET    /rails/active_storage/representations/:signed_blob_id/:variation_key/*filename(.:format) active_storage/representations#show
       rails_disk_service GET    /rails/active_storage/disk/:encoded_key/*filename(.:format)                              active_storage/disk#show
update_rails_disk_service PUT    /rails/active_storage/disk/:encoded_token(.:format)                                      active_storage/disk#update
     rails_direct_uploads POST   /rails/active_storage/direct_uploads(.:format)                                           active_storage/direct_uploads#create
```

Aula 15 - Symbols x Strings e brincando com Active Record

:texto x "texto"



```
C:\Sites\exemplos>rails c
Loading development environment (Rails 5.2.3)
irb(main):001:0> "rails".class
=> String
irb(main):002:0> "rails".object_id
=> 64345300
irb(main):003:0> "rails".object_id
=> 64528060
irb(main):004:0> "rails".object_id
=> 64688600
irb(main):005:0> :rails.class
=> Symbol
irb(main):006:0> :rails.object_id
=> 1786268
irb(main):007:0> :rails.object_id
=> 1786268
irb(main):008:0> :rails.object_id
=> 1786268
```

Símbolo não é repetido na memória.
String, sempre que é invocada é criada uma nova instância em memória.

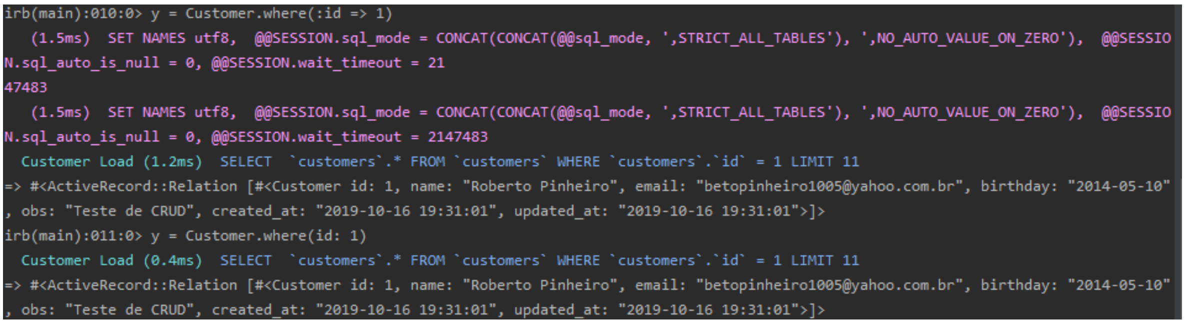
A diferença básica está na performance (relacionada ao uso de memória).

Active Record

Pesquisando no BD

rails c

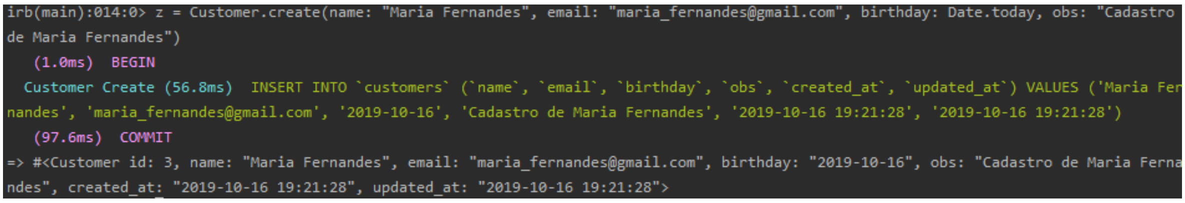
y = Customer.where(id: 1) ou y = Customer.where(:id => 1)



```
irb(main):010:0> y = Customer.where(:id => 1)
   (1.5ms)  SET NAMES utf8,  @@SESSION.sql_mode = CONCAT(CONCAT(@@sql_mode, ',STRICT_ALL_TABLES'), ',NO_AUTO_VALUE_ON_ZERO'),  @@SESSIO
N.sql_auto_is_null = 0, @@SESSION.wait_timeout = 21
47483
   (1.5ms)  SET NAMES utf8,  @@SESSION.sql_mode = CONCAT(CONCAT(@@sql_mode, ',STRICT_ALL_TABLES'), ',NO_AUTO_VALUE_ON_ZERO'),  @@SESSIO
N.sql_auto_is_null = 0, @@SESSION.wait_timeout = 2147483
  Customer Load (1.2ms)  SELECT  `customers`.* FROM `customers` WHERE `customers`.`id` = 1 LIMIT 11
=> #<ActiveRecord::Relation [#<Customer id: 1, name: "Roberto Pinheiro", email: "betopinheiro1005@yahoo.com.br", birthday: "2014-05-10"
, obs: "Teste de CRUD", created_at: "2019-10-16 19:31:01", updated_at: "2019-10-16 19:31:01">]>
irb(main):011:0> y = Customer.where(id: 1)
  Customer Load (0.4ms)  SELECT  `customers`.* FROM `customers` WHERE `customers`.`id` = 1 LIMIT 11
=> #<ActiveRecord::Relation [#<Customer id: 1, name: "Roberto Pinheiro", email: "betopinheiro1005@yahoo.com.br", birthday: "2014-05-10"
, obs: "Teste de CRUD", created_at: "2019-10-16 19:31:01", updated_at: "2019-10-16 19:31:01">]>
```

Criando e persistindo no BD

z = Customer.create(name: "Maria Fernandes", email: "maria_fernandes@gmail.com", birthday: Date.today, obs: "Cadastro de Maria Fernandes")



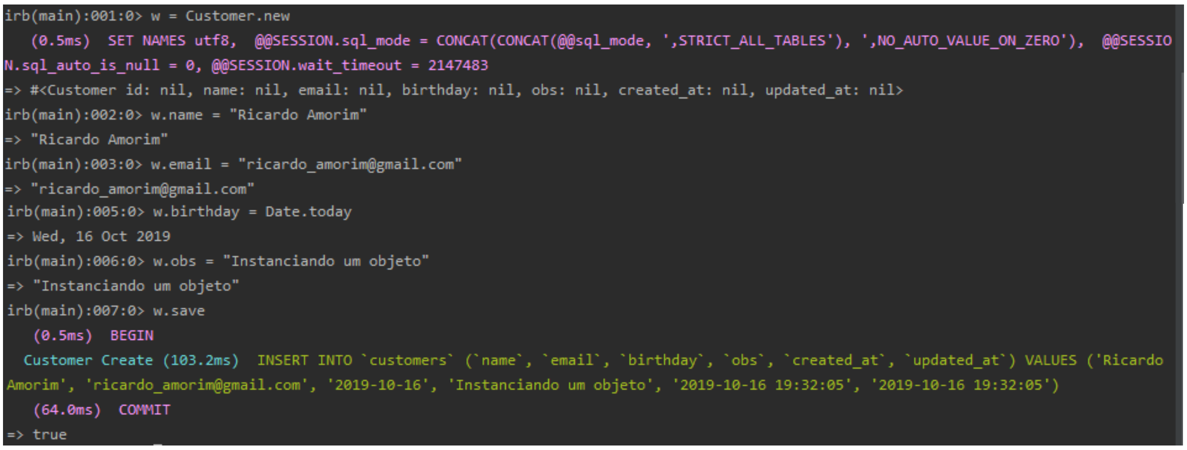
```
irb(main):014:0> z = Customer.create(name: "Maria Fernandes", email: "maria_fernandes@gmail.com", birthday: Date.today, obs: "Cadastro
de Maria Fernandes")
   (1.0ms)  BEGIN
  Customer Create (56.8ms)  INSERT INTO `customers` (`name`, `email`, `birthday`, `obs`, `created_at`, `updated_at`) VALUES ('Maria Fer
nandes', 'maria_fernandes@gmail.com', '2019-10-16', 'Cadastro de Maria Fernandes', '2019-10-16 19:21:28', '2019-10-16 19:21:28')
   (97.6ms)  COMMIT
=> #<Customer id: 3, name: "Maria Fernandes", email: "maria_fernandes@gmail.com", birthday: "2019-10-16", obs: "Cadastro de Maria Ferna
ndes", created_at: "2019-10-16 19:21:28", updated_at: "2019-10-16 19:21:28">
```

Instanciando o objeto

w = Customer.new

w.name = "Ricardo Amorim"
w.email = "ricardo_amorim@gmail.com"
w.birthday = "Date.today"
w.obs = "Instanciando um objeto"

w.save



```
irb(main):001:0> w = Customer.new
   (0.5ms)  SET NAMES utf8,  @@SESSION.sql_mode = CONCAT(CONCAT(@@sql_mode, ',STRICT_ALL_TABLES'), ',NO_AUTO_VALUE_ON_ZERO'),  @@SESSIO
N.sql_auto_is_null = 0, @@SESSION.wait_timeout = 2147483
=> #<Customer id: nil, name: nil, email: nil, birthday: nil, obs: nil, created_at: nil, updated_at: nil>
irb(main):002:0> w.name = "Ricardo Amorim"
=> "Ricardo Amorim"
irb(main):003:0> w.email = "ricardo_amorim@gmail.com"
=> "ricardo_amorim@gmail.com"
irb(main):005:0> w.birthday = Date.today
=> Wed, 16 Oct 2019
irb(main):006:0> w.obs = "Instanciando um objeto"
=> "Instanciando um objeto"
irb(main):007:0> w.save
   (0.5ms)  BEGIN
  Customer Create (103.2ms)  INSERT INTO `customers` (`name`, `email`, `birthday`, `obs`, `created_at`, `updated_at`) VALUES ('Ricardo
Amorim', 'ricardo_amorim@gmail.com', '2019-10-16', 'Instanciando um objeto', '2019-10-16 19:32:05', '2019-10-16 19:32:05')
   (64.0ms)  COMMIT
=> true
```

Aula 16 - Controllers, Filtros e Partials

Filtros

Filtros são métodos que são rodados antes (before_action), durante (around_action) ou depois (after_action) de uma ação de um controller.

before_action x before_filter

São a mesma coisa.

Antes da versão 4 do Ruby usava-se before_filter. Da versão 4 em diante usa-se before_action.

Exemplo:

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

 def show
 end

 def edit
 end

 def update
  respond_to do |format|
   if @customer.update(customer_params)
    format.html { redirect_to @customer, notice: 'Customer was successfully updated.' }
    format.json { render :show, status: :ok, location: @customer }
   else
    format.html { render :edit }
    format.json { render json: @customer.errors, status: :unprocessable_entity }
   end
  end
 end

 def destroy
  @customer.destroy
  respond_to do |format|
   format.html { redirect_to customers_url, notice: 'Customer was successfully destroyed.' }
   format.json { head :no_content }
  end
 end
```

```
  private

    # Use callbacks to share common setup or constraints between actions.
    def set_customer
      @customer = Customer.find(params[:id])
    end
```

Partials

É um arquivo com a extensão .html.erb que começa com underline(_) e que pode ser compartilhado entre outros arquivos.

Exemplo:

_form.html.erb

Aula 17 – Helpers e Params

form(method = "post", action = script_name, enctype = "application/x-www-form-urlencoded")
public
Generate a Form element as a string.

method should be either “get” or “post”, and defaults to the latter. action defaults to the current CGI script name. enctype defaults to “application/x-www-form-urlencoded”.

Alternatively, the attributes can be specified as a hash.

See also #multipart_form() for forms that include file uploads.

```
form{ "string" }
  # <FORM METHOD="post" ENCTYPE="application/x-www-form-urlencoded">string</FORM>

form("get") { "string" }
  # <FORM METHOD="get" ENCTYPE="application/x-www-form-urlencoded">string</FORM>

form("get", "url") { "string" }
  # <FORM METHOD="get" ACTION="url" ENCTYPE="application/x-www-form-urlencoded">string</FORM>

form("METHOD" => "post", "ENCTYPE" => "enctype") { "string" }
  # <FORM METHOD="post" ENCTYPE="enctype">string</FORM>
```

text_field(name = "", value = nil, size = 40, maxlength = nil) public

Generate a text field Input element, as a String.

name is the name of the input field. value is its initial value. size is the size of the input area. maxlength is the maximum length of input accepted.

Alternatively, the attributes can be specified as a hash.

```
text_field("name")
  # <INPUT TYPE="text" NAME="name" SIZE="40">

text_field("name", "value")
  # <INPUT TYPE="text" NAME="name" VALUE="value" SIZE="40">

text_field("name", "value", 80)
  # <INPUT TYPE="text" NAME="name" VALUE="value" SIZE="80">

text_field("name", "value", 80, 200)
  # <INPUT TYPE="text" NAME="name" VALUE="value" SIZE="80" MAXLENGTH="200">

text_field("NAME" => "name", "VALUE" => "value")
  # <INPUT TYPE="text" NAME="name" VALUE="value">
```

text_field(name = "", value = nil, size = 40, maxlength = nil) public

Generate a text field Input element, as a String.

name is the name of the input field. value is its initial value. size is the size of the input area. maxlength is the maximum length of input accepted.

Alternatively, the attributes can be specified as a hash.

```
text_field("name")
  # <INPUT TYPE="text" NAME="name" SIZE="40">

text_field("name", "value")
  # <INPUT TYPE="text" NAME="name" VALUE="value" SIZE="40">

text_field("name", "value", 80)
  # <INPUT TYPE="text" NAME="name" VALUE="value" SIZE="80">

text_field("name", "value", 80, 200)
  # <INPUT TYPE="text" NAME="name" VALUE="value" SIZE="80" MAXLENGTH="200">

text_field("NAME" => "name", "VALUE" => "value")
  # <INPUT TYPE="text" NAME="name" VALUE="value">
```

submit(value = nil, name = nil) public

Generate a submit button Input element, as a String.

value is the text to display on the button. name is the name of the input.

Alternatively, the attributes can be specified as a hash.

```
submit
  # <INPUT TYPE="submit">

submit("ok")
  # <INPUT TYPE="submit" VALUE="ok">

submit("ok", "button1")
  # <INPUT TYPE="submit" VALUE="ok" NAME="button1">

submit("VALUE" => "ok", "NAME" => "button1", "ID" => "foo")
  # <INPUT TYPE="submit" VALUE="ok" NAME="button1" ID="foo">
```

Params

Exemplo:

```
  private
    # Use callbacks to share common setup or constraints between actions.
    def set_customer
      @customer = Customer.find(params[:id])
    end

    # Never trust parameters from the scary internet, only allow the white list through.
    def customer_params
      params.require(:customer).permit(:name, :email, :birthday, :obs)
    end
```

Aula 18 – Internationalization

i18n

config/application.rb

```
require_relative 'boot'

require 'rails/all'

# Require the gems listed in Gemfile, including any gems
# you've limited to :test, :development, or :production.
Bundler.require(*Rails.groups)

module Exemplos
  class Application < Rails::Application
    # Initialize configuration defaults for originally generated Rails version.
    config.load_defaults 5.2

    # Settings in config/environments/* take precedence over those specified here.
    # Application configuration can go into files in config/initializers
    # -- all .rb files in that directory are automatically loaded after loading
    # the framework and any gems in your application.
    config.i18n.default_locale = "pt-BR"
  end
end
```

Como usar?

translate # Traduzir textos
localize # Localizar data/hora

i18n.t # Traduzir textos
i18n.l # Localizar data/hora

Nas views:

t # Traduzir textos
l # Localizar data/hora

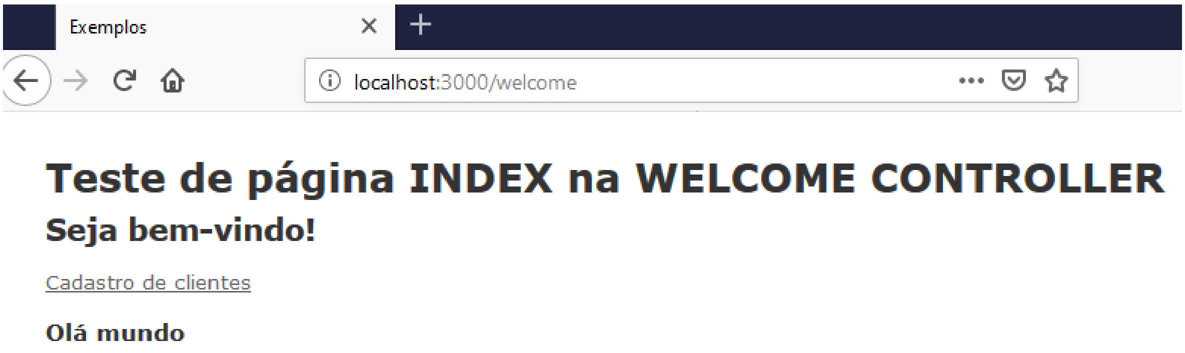
config/locales/pt-br.yml

```
pt-BR:
  hello: "Olá mundo"
```

- Reinicie o servidor.

app/views/welcome/index.html.erb

```
<h1>Teste de página INDEX na WELCOME CONTROLLER</h1>
<h2><%= @message %></h2>
<%= link_to "Cadastro de clientes", customers_path %>

<h3><%= t('hello') %></h3>
```



Teste de página INDEX na WELCOME CONTROLLER

Seja bem-vindo!

Cadastro de clientes

Olá mundo

- Acesse a URL:

https://github.com/svenfuchs/rails-i18n/tree/master/rails/locale

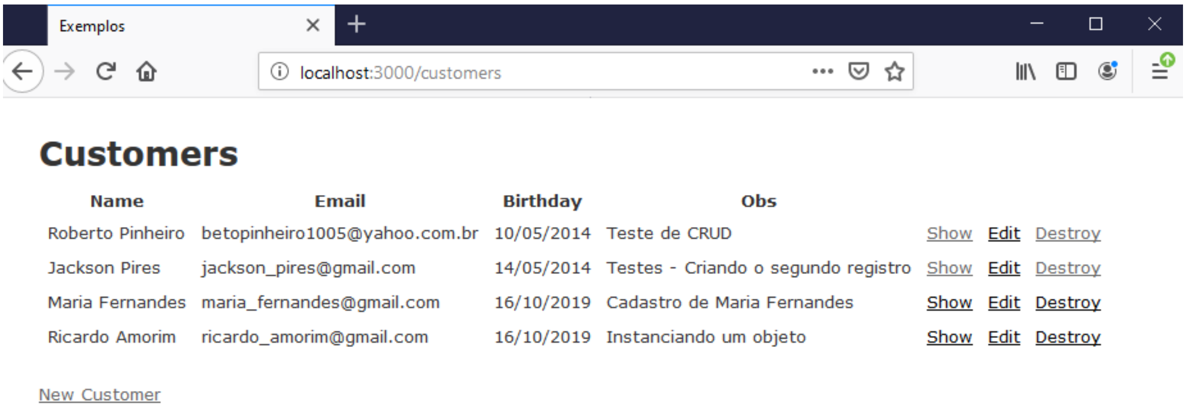
E copie o conteúdo no arquivo pt-BR.yml

config/locales/pt-br_rails.rb

```
pt-BR:
  activerecord:
    errors:
      messages:
        record_invalid: 'A validação falhou: %{errors}'
        restrict_dependent_destroy:
          has_one: Não é possível excluir o registro pois existe um %{record} dependente
          has_many: Não é possível excluir o registro pois existem %{record} dependentes
  date:
    abbr_day_names:
    - dom
    - seg
    - ter
    - qua
    - qui
    - sex
    - sáb
    abbr_month_names:
    -
    - jan
    - fev
    - mar
    - abr
    - mai
    - jun
    - jul
    - ago
    - set
    - out
    - nov
    - dez
    day_names:
    - domingo
    - segunda-feira
    - terça-feira
    - quarta-feira
    - quinta-feira
    - sexta-feira
    - sábado
    formats:
      default: "%d/%m/%Y"
      long: "%d de %B de %Y"
      short: "%d de %B"
    month_names:
    -
    - janeiro
    - fevereiro
    - março
    - abril
    - maio
    - junho
    - julho
    - agosto
    - setembro
    - outubro
    - novembro
    - dezembro
    order:
    - :day
    - :month
    - :year
  datetime:
    distance_in_words:
      about_x_hours:
        one: aproximadamente 1 hora
        other: aproximadamente %{count} horas
      about_x_months:
        one: aproximadamente 1 mês
        other: aproximadamente %{count} meses
      about_x_years:
        one: aproximadamente 1 ano
        other: aproximadamente %{count} anos
      almost_x_years:
        one: quase 1 ano
        other: quase %{count} anos
      half_a_minute: meio minuto
      less_than_x_seconds:
        one: menos de 1 segundo
        other: menos de %{count} segundos
      less_than_x_minutes:
        one: menos de um minuto
        other: menos de %{count} minutos
      over_x_years:
        one: mais de 1 ano
        other: mais de %{count} anos
      x_seconds:
        one: 1 segundo
        other: "%{count} segundos"
      x_minutes:
        one: 1 minuto
        other: "%{count} minutos"
      x_days:
        one: 1 dia
        other: "%{count} dias"
      x_months:
        one: 1 mês
        other: "%{count} meses"
      x_years:
        one: 1 ano
        other: "%{count} anos"
```

```yaml
  prompts:
    second: Segundo
    minute: Minuto
    hour: Hora
    day: Dia
    month: Mês
    year: Ano
errors:
  format: "%{attribute} %{message}"
  messages:
    accepted: deve ser aceito
    blank: não pode ficar em branco
    confirmation: não é igual a %{attribute}
    empty: não pode ficar vazio
    equal_to: deve ser igual a %{count}
    even: deve ser par
    exclusion: não está disponível
    greater_than: deve ser maior que %{count}
    greater_than_or_equal_to: deve ser maior ou igual a %{count}
    inclusion: não está incluído na lista
    invalid: não é válido
    less_than: deve ser menor que %{count}
    less_than_or_equal_to: deve ser menor ou igual a %{count}
    model_invalid: 'A validação falhou: %{errors}'
    not_a_number: não é um número
    not_an_integer: não é um número inteiro
    odd: deve ser ímpar
    other_than: deve ser diferente de %{count}
    present: deve ficar em branco
    required: é obrigatório(a)
    taken: já está em uso
    too_long:
      one: 'é muito longo (máximo: 1 caracter)'
      other: 'é muito longo (máximo: %{count} caracteres)'
    too_short:
      one: 'é muito curto (mínimo: 1 caracter)'
      other: 'é muito curto (mínimo: %{count} caracteres)'
    wrong_length:
      one: não possui o tamanho esperado (1 caracter)
      other: não possui o tamanho esperado (%{count} caracteres)
  template:
    body: 'Por favor, verifique o(s) seguinte(s) campo(s):'
    header:
      one: 'Não foi possível gravar %{model}: 1 erro'
      other: 'Não foi possível gravar %{model}: %{count} erros'
helpers:
  select:
    prompt: Por favor selecione
  submit:
    create: Criar %{model}
    submit: Salvar %{model}
    update: Atualizar %{model}
```

```
number:
  currency:
    format:
      delimiter: "."
      format: "%u %n"
      precision: 2
      separator: ","
      significant: false
      strip_insignificant_zeros: false
      unit: R$
  format:
    delimiter: "."
    precision: 3
    separator: ","
    significant: false
    strip_insignificant_zeros: false
  human:
    decimal_units:
      format: "%n %u"
      units:
        billion:
          one: bilhão
          other: bilhões
        million:
          one: milhão
          other: milhões
        quadrillion:
          one: quatrilhão
          other: quatrilhões
        thousand: mil
        trillion:
          one: trilhão
          other: trilhões
        unit: ''
    format:
      delimiter: ''
      precision: 3
      significant: true
      strip_insignificant_zeros: true
    storage_units:
      format: "%n %u"
      units:
        byte:
          one: Byte
          other: Bytes
        eb: EB
        gb: GB
        kb: KB
        mb: MB
        pb: PB
        tb: TB
  percentage:
   format:
    delimiter: "."
    format: "%n%"
  precision:
   format:
    delimiter: "."
 support:
  array:
   last_word_connector: " e "
   two_words_connector: " e "
   words_connector: ", "
 time:
  am: ''
  formats:
   default: "%a, %d de %B de %Y, %H:%M:%S %z"
   long: "%d de %B de %Y, %H:%M"
   short: "%d de %B, %H:%M"
  pm: ''
```



Customers

Name	Email	Birthday	Obs			
Roberto Pinheiro	betopinheiro1005@yahoo.com.br	10/05/2014	Teste de CRUD	Show	Edit	Destroy
Jackson Pires	jackson_pires@gmail.com	14/05/2014	Testes - Criando o segundo registro	Show	Edit	Destroy
Maria Fernandes	maria_fernandes@gmail.com	16/10/2019	Cadastro de Maria Fernandes	Show	Edit	Destroy
Ricardo Amorim	ricardo_amorim@gmail.com	16/10/2019	Instanciando um objeto	Show	Edit	Destroy

New Customer

Aula 19 - Twitter Bootstrap e i18n para Active Record

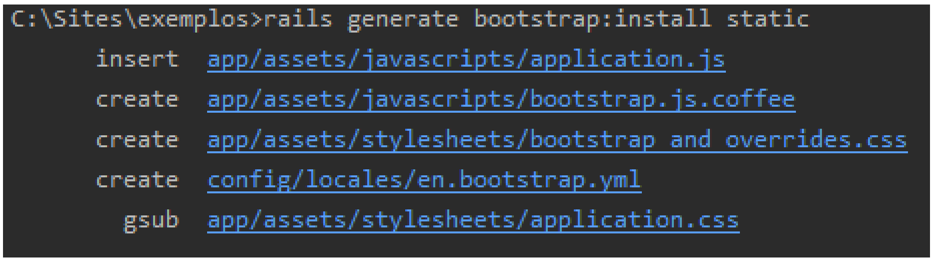
Gemfile

```
source 'https://rubygems.org'
git_source(:github) { |repo| "https://github.com/#{repo}.git" }

ruby '2.5.1'

gem 'pry', '~> 0.12.2'
gem 'rb-readline', '~> 0.5.3'

# Twitter Bootstrap for Rails 6, 5 and 4 Asset Pipeline
gem 'twitter-bootstrap-rails', :git => 'git://github.com/seyhunak/twitter-bootstrap-rails.git'

# Bundle edge Rails instead: gem 'rails', github: 'rails/rails'
gem 'rails', '~> 5.2.1'
# Use mysql as the database for Active Record
gem 'mysql2', '>= 0.4.4', '< 0.6.0'
# Use Puma as the app server
gem 'puma', '~> 3.11'
# Use SCSS for stylesheets
gem 'sass-rails', '~> 5.0'
# Use Uglifier as compressor for JavaScript assets
gem 'uglifier', '>= 1.3.0'
# See https://github.com/rails/execjs#readme for more supported runtimes
gem 'duktape'
# Use CoffeeScript for .coffee assets and views
gem 'coffee-rails', '~> 4.2'
# Turbolinks makes navigating your web application faster. Read more:
https://github.com/turbolinks/turbolinks
gem 'turbolinks', '~> 5'
# Build JSON APIs with ease. Read more: https://github.com/rails/jbuilder
gem 'jbuilder', '~> 2.5'
# Use Redis adapter to run Action Cable in production
# gem 'redis', '~> 4.0'
# Use ActiveModel has_secure_password
# gem 'bcrypt', '~> 3.1.7'

# Use ActiveStorage variant
# gem 'mini_magick', '~> 4.8'

# Use Capistrano for deployment
# gem 'capistrano-rails', group: :development

# Reduces boot times through caching; required in config/boot.rb
gem 'bootsnap', '>= 1.1.0', require: false

group :development, :test do
```

```
  # Call 'byebug' anywhere in the code to stop execution and get a debugger console
  gem 'byebug', platforms: [:mri, :mingw, :x64_mingw]
end

group :development do
  # Access an interactive console on exception pages or by calling 'console' anywhere in the code.
  gem 'web-console', '>= 3.3.0'
end

group :test do
  # Adds support for Capybara system testing and selenium driver
  gem 'capybara', '>= 2.15'
  gem 'selenium-webdriver'
  # Easy installation and use of chromedriver to run system tests with Chrome
  gem 'chromedriver-helper'
end

# Windows does not include zoneinfo files, so bundle the tzinfo-data gem
gem 'tzinfo-data', platforms: [:mingw, :mswin, :x64_mingw, :jruby]
```

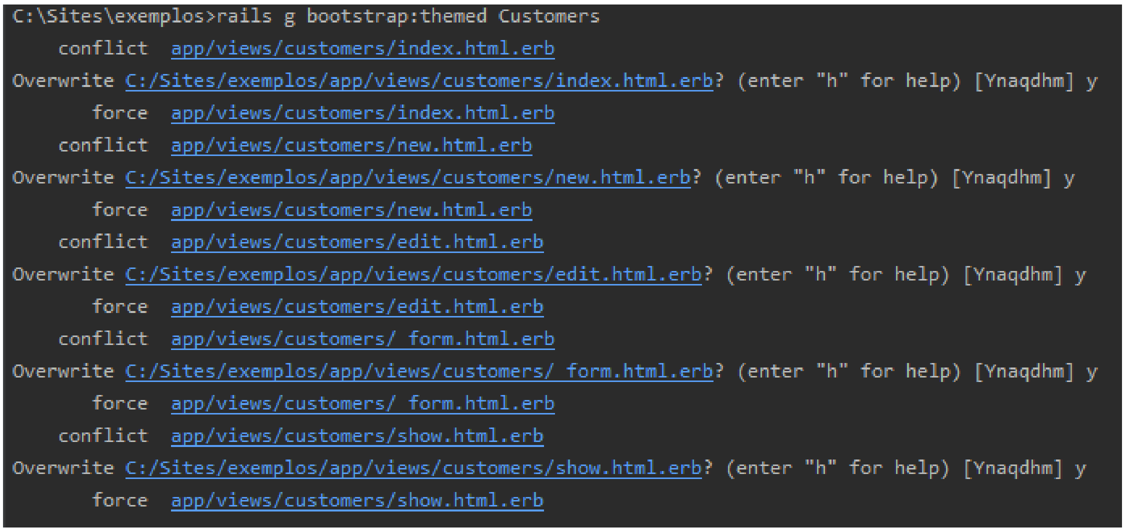
bundle install

rails generate bootstrap:install static



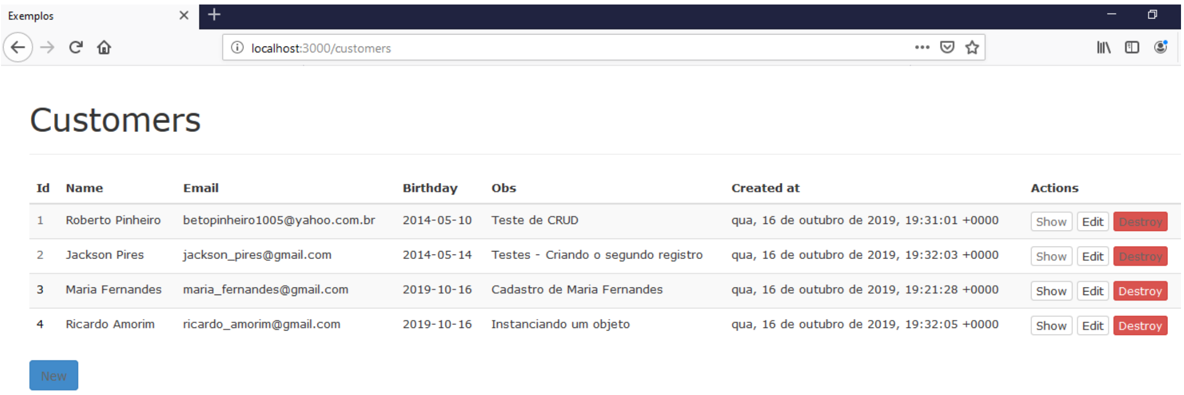
```
C:\Sites\exemplos>rails generate bootstrap:install static
      insert  app/assets/javascripts/application.js
      create  app/assets/javascripts/bootstrap.js.coffee
      create  app/assets/stylesheets/bootstrap_and_overrides.css
      create  config/locales/en.bootstrap.yml
        gsub  app/assets/stylesheets/application.css
```

rails g bootstrap:themed Customers



```
C:\Sites\exemplos>rails g bootstrap:themed Customers
    conflict  app/views/customers/index.html.erb
Overwrite C:/Sites/exemplos/app/views/customers/index.html.erb? (enter "h" for help) [Ynaqdhm] y
       force  app/views/customers/index.html.erb
    conflict  app/views/customers/new.html.erb
Overwrite C:/Sites/exemplos/app/views/customers/new.html.erb? (enter "h" for help) [Ynaqdhm] y
       force  app/views/customers/new.html.erb
    conflict  app/views/customers/edit.html.erb
Overwrite C:/Sites/exemplos/app/views/customers/edit.html.erb? (enter "h" for help) [Ynaqdhm] y
       force  app/views/customers/edit.html.erb
    conflict  app/views/customers/_form.html.erb
Overwrite C:/Sites/exemplos/app/views/customers/_form.html.erb? (enter "h" for help) [Ynaqdhm] y
       force  app/views/customers/_form.html.erb
    conflict  app/views/customers/show.html.erb
Overwrite C:/Sites/exemplos/app/views/customers/show.html.erb? (enter "h" for help) [Ynaqdhm] y
       force  app/views/customers/show.html.erb
```

- Reinicie o servidor para que as alterações tenham efeito.



Customers

Id	Name	Email	Birthday	Obs	Created at	Actions
1	Roberto Pinheiro	betopinheiro1005@yahoo.com.br	2014-05-10	Teste de CRUD	qua, 16 de outubro de 2019, 19:31:01 +0000	Show Edit Destroy
2	Jackson Pires	jackson_pires@gmail.com	2014-05-14	Testes - Criando o segundo registro	qua, 16 de outubro de 2019, 19:32:03 +0000	Show Edit Destroy
3	Maria Fernandes	maria_fernandes@gmail.com	2019-10-16	Cadastro de Maria Fernandes	qua, 16 de outubro de 2019, 19:21:28 +0000	Show Edit Destroy
4	Ricardo Amorim	ricardo_amorim@gmail.com	2019-10-16	Instanciando um objeto	qua, 16 de outubro de 2019, 19:32:05 +0000	Show Edit Destroy

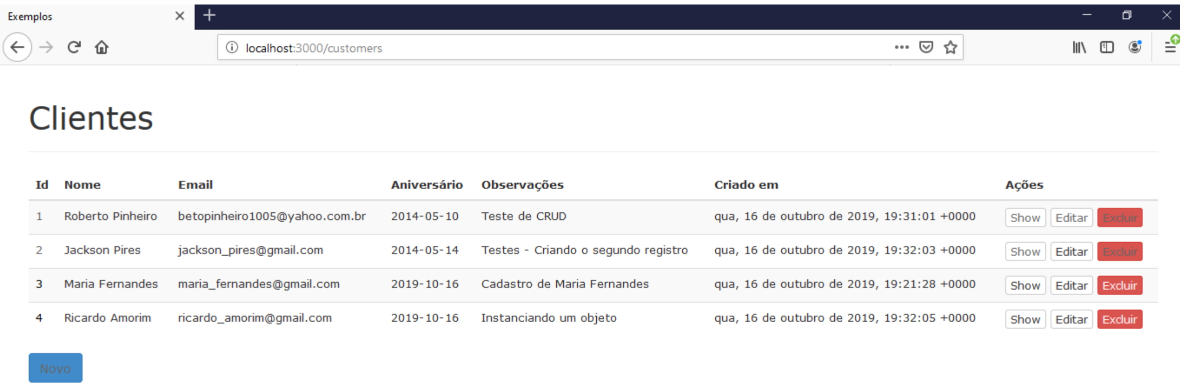
Traduções

config/locales/pt-br_models.yml

```
pt-BR:
  activerecord:
    models:
      customer: Cliente
    attributes:
      customer:
        name: Nome
        email: Email
        birthday: Aniversário
        obs: Observações
        created_at: Criado em
```

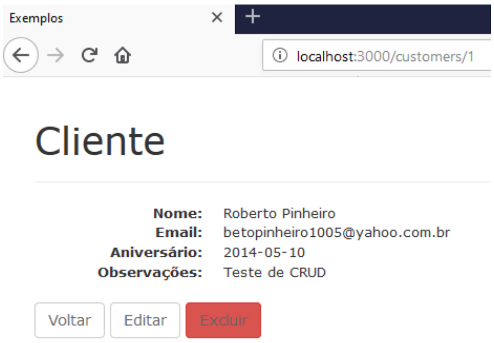
config/locales/pt-br_bootstrap.yml

```
# Sample localization file for English. Add more files in this directory for other locales.
# See https://github.com/svenfuchs/rails-i18n/tree/master/rails%2Flocale for starting points.

pt-BR:
  breadcrumbs:
    application:
      root: "Index"
    pages:
      pages: "Páginas"
  helpers:
    actions: "Ações"
    links:
      back: "Voltar"
      cancel: "Cancelar"
      confirm: "Tem certeza?"
      destroy: "Excluir"
      new: "Novo"
      edit: "Editar"
    titles:
      edit: "Editar %{model}"
      save: "Salvar %{model}"
      new: "Novo %{model}"
      delete: "Apagar %{model}"
```



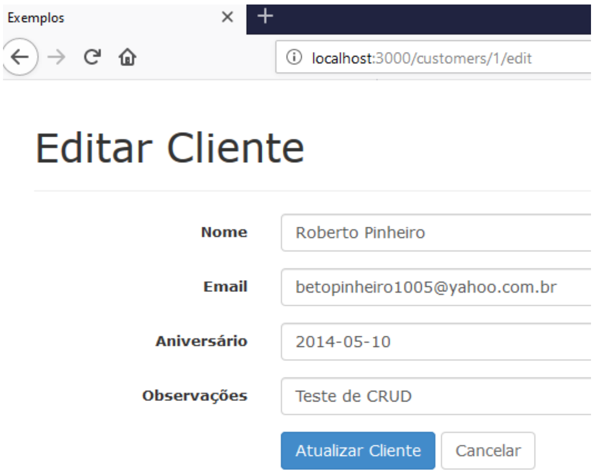
Clientes

Id	Nome	Email	Aniversário	Observações	Criado em	Ações
1	Roberto Pinheiro	betopinheiro1005@yahoo.com.br	2014-05-10	Teste de CRUD	qua, 16 de outubro de 2019, 19:31:01 +0000	Show Editar Excluir
2	Jackson Pires	jackson_pires@gmail.com	2014-05-14	Testes - Criando o segundo registro	qua, 16 de outubro de 2019, 19:32:03 +0000	Show Editar Excluir
3	Maria Fernandes	maria_fernandes@gmail.com	2019-10-16	Cadastro de Maria Fernandes	qua, 16 de outubro de 2019, 19:21:28 +0000	Show Editar Excluir
4	Ricardo Amorim	ricardo_amorim@gmail.com	2019-10-16	Instanciando um objeto	qua, 16 de outubro de 2019, 19:32:05 +0000	Show Editar Excluir



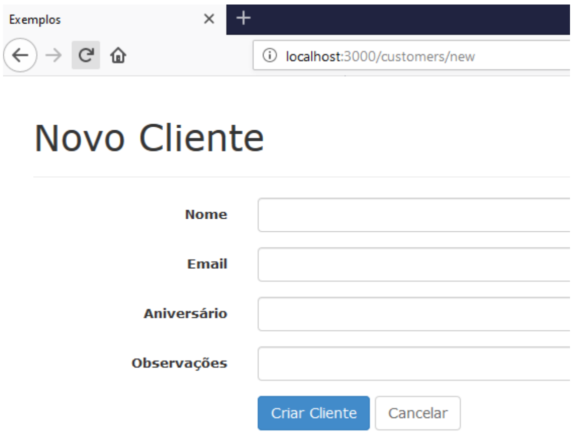
Cliente

Nome: Roberto Pinheiro
Email: betopinheiro1005@yahoo.com.br
Aniversário: 2014-05-10
Observações: Teste de CRUD



Editar Cliente

Nome Roberto Pinheiro
Email betopinheiro1005@yahoo.com.br
Aniversário 2014-05-10
Observações Teste de CRUD



Novo Cliente

Nome
Email
Aniversário
Observações