

Ruby on Rails



Pop Alexandru
Pavel Razvan
Calculatoare 4

Ruby

- Aparut in 1995
- Limbaj de programare
- Este gratuit si open source
- Foloseste librarii numite “gem”-uri
- Foloseste doar programarea orientata pe obiecte
- Ver 2.2.2 (13 aprilie 2015)

Rails

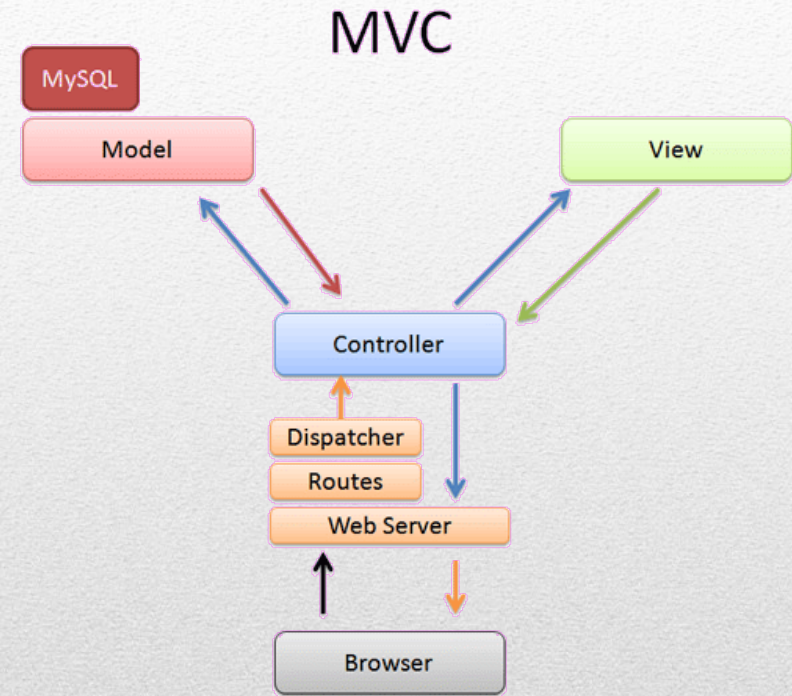
- Aparut in 2004
- Framework web bazat pe ruby
- Este gratuit si open source
- Este defapt un gem pentru crearea unui server
- Foloseste arhitectura Model-View-Controller
- Ver 4.2.1 (19 martie 2015)

Ruby **VS** Rails

- Sintaxa este foarte similara cu python si perl, ajutand la scrierea codului rapid si eficient
- Contine un garbage collector
- Contine by default un irb(Interactive Ruby Shell) in care se poate scrie cod ruby si se porneste simplu, tastand “irb” in consola/terminal
- Fiindca orice variabila sau constanta e un obiect, ruby permite diferite apeluri direct pe numere care au clasa Integer(de exemplu “2.to_f”)
- Variabilele care incep cu @ sunt variabile de instanta, iar cele cu @@ sunt variabile de clasa
- Nu exista notiunea de caracter, deci “ex” si ‘ex’ sunt aproape identice
- Nu este strongly-typed deci nu avem nevoie sa instantiem variabile cu o anumita clasa

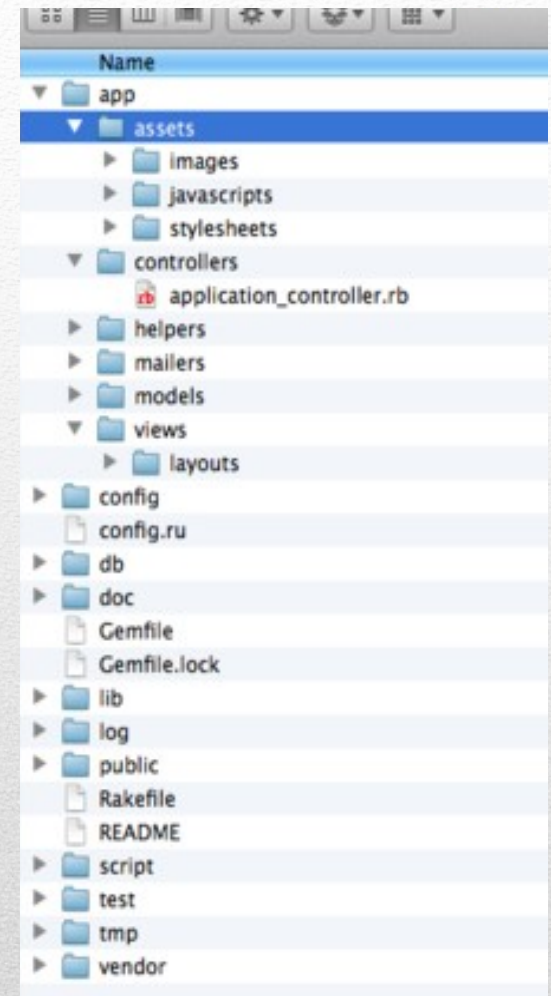
Introducere Ruby

- Aplicatia este apelata prin intermediul browserului cu un url
- Web Serverul(Webrick, Unicorn, Thin, Passenger, Puma, Mongrel) primeste apelul si verifica daca exista ruta valida care are atasat un controller si o metoda din controller
- Controllerul aduce informatii de la baza de date(MySQL, MongoDB, PostgreSQL) si le pune in variabile de instanta
- Introduce datele in fisierul din view(html, haml care poate contine cod javascript, coffescript, css, scss, sass) si le trimite browserului, care randeaza pe ecran htmlul final



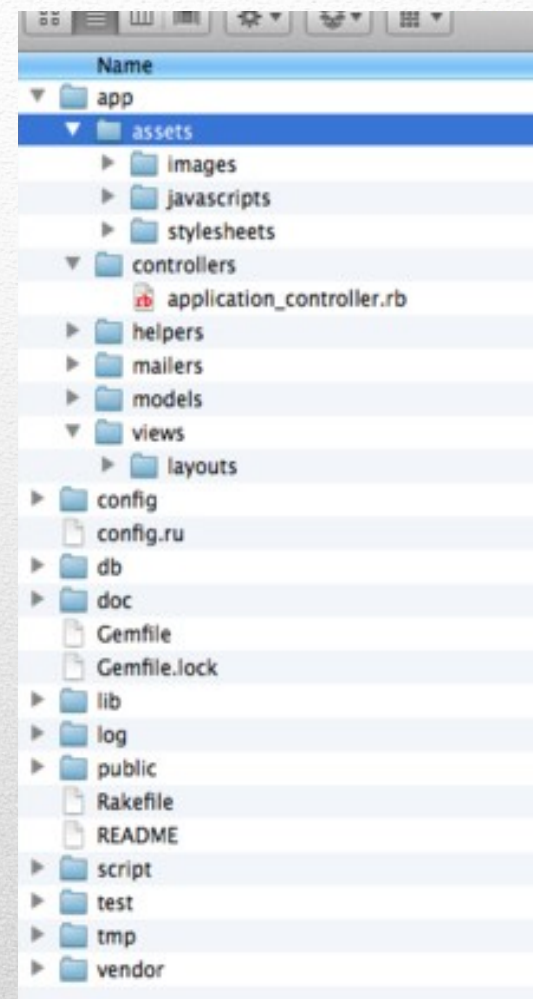
Introducere Rails

- Assets: fisiere media, javascript si css care sunt inserate in html
- Controllers: fisiere care contin clase si metode pentru controller(application_controller.rb este principalul controller de la care vor fi mostenite celelalte controllere); acestea pot fi raspunde cu mai multe tipuri de formate: html, json, xml, pdf, xls
- Helpers: clase si metode care sunt folosite in view pentru a reduce repetitiile si a face codul cat mai mic si concis
- Mailers: clase si metode care mostenesc modulul ActionMailer(este un gem instalat implicit) si contin logica trimiterii emailurilor

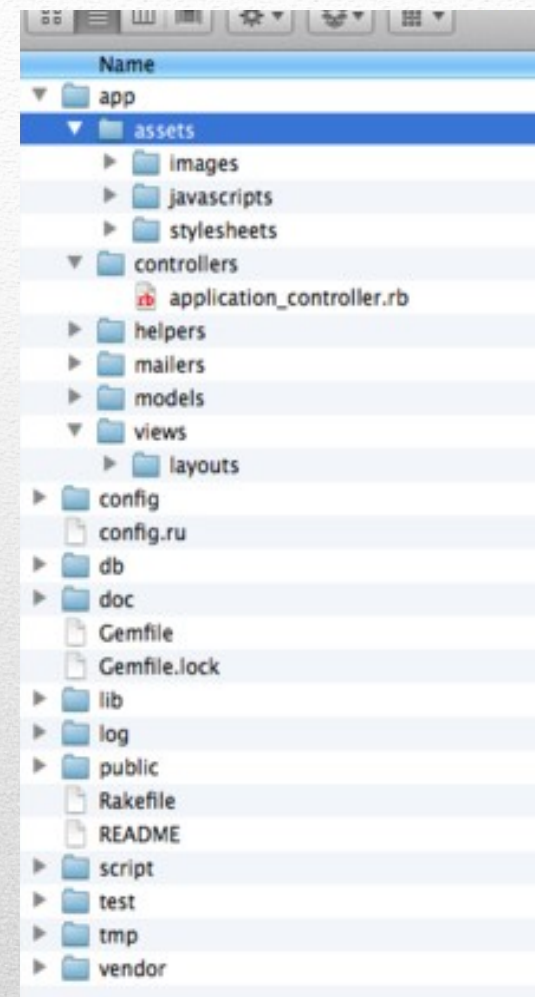


Structura directoarelor

- Models: clase si metode care mostenesc de obicei modulul ActiveRecord(este un gem instalat implicit pentru MySQL si PostgreSQL) si contin logica accesarii bazei de date si manipulare a datelor
- Views: fisiere html.erb(extensia prin care putem introduce cod ruby in html), haml
- Views/layouts: fisiere template pentru a ingloba alte fisiere html, folosite cel mai des pentru header si footer; astfel celelalte fisiere din view vor contine doar date referitoare la logica controllerului in care se afla, evitand repetitia tagurilor de <head></head>, <body></body>, header si footer.
- Config: fisiere rb, yaml sau yml care contin diferite configurari precum date de autentificare si conectare la baza de date, la servicii externe , configurarea diferitelor medii de dezvoltare(development, staging, test, production), constante, configurari pentru gemuri
- Db: fisiere ruby care permit modificarea structurii bazei de date si care sunt executate cu comanda “rake db:migrate”

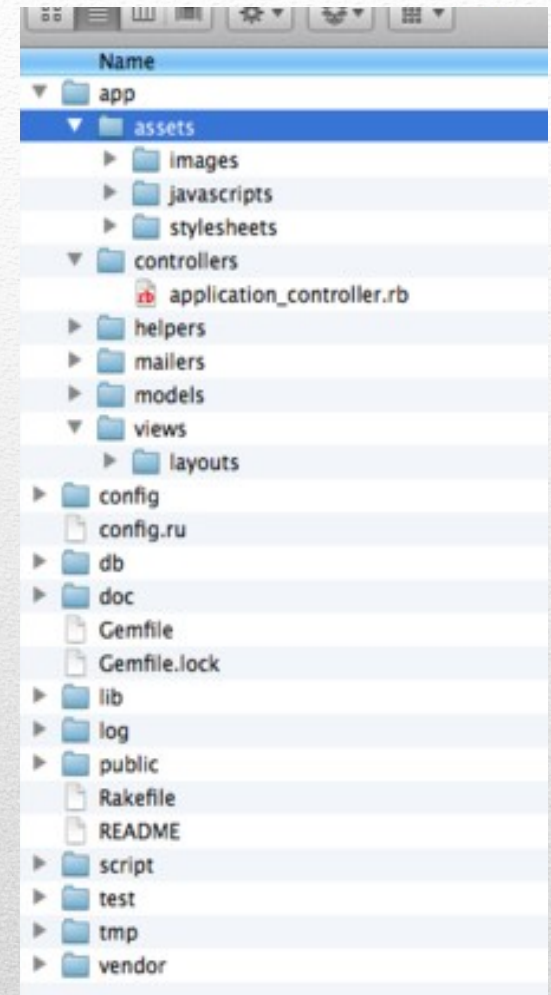


- Assets: fisiere media, javascript si css care sunt inserate in html
- Controllers: fisiere care contin clase si metode pentru controller(application_controller.rb este principalul controller de la care vor fi mostenite celelalte controllere); acestea pot fi raspunde cu mai multe tipuri de formate: html, json, xml, pdf, xls
- Helpers: clase si metode care sunt folosite in view pentru a reduce repetitiile si a face codul cat mai mic si concis
- Mailers: clase si metode care mostenesc modulul ActionMailer(este un gem instalat implicit) si contin logica trimiterii emailurilor



Structura directoarelor

- Doc: fișiere pentru documentația aplicației
- Gemfile și Gemfile.lock: conțin gemurile importate în aplicație și instalate pe sistemul de calcul (folosind comanda “bundle install”) de obicei din surse externe (cele mai des întâlnite sunt rubygems.org și github.com)
- Lib, script: librării, scripturi interne
- Public: fișiere statice care pot fi accesate fără arhitectura MVC și care nu au nevoie de rute predefinite (fișierul de eroare 404.html)
- Test: fișiere ruby pentru testarea tuturor metodelor folosite în aplicație care folosesc mediul testing și o altă bază de date, care nu conține date, doar structura celei originale
- Log: fișiere cu extensia log, care conțin date de logare
- Tmp: fișiere temporare precum cele cu extensia pid, care conțin date referitoare la procesele pornite în aplicație



```
n@razvan-HP:~/ruby$ rails new web_tech
create
create  README.rdoc
create  Rakefile
create  config.ru
create  .gitignore
create  Gemfile
create  app
create  app/assets/javascripts/application.js
create  app/assets/stylesheets/application.css
create  app/controllers/application_controller.rb
create  app/helpers/application_helper.rb
create  app/views/layouts/application.html.erb
create  app/assets/images/.keep
create  app/mailers/.keep
create  app/models/.keep
create  app/controllers/concerns/.keep
create  app/models/concerns/.keep
create  bin
create  bin/bundle
create  bin/rails
create  bin/rake
create  bin/setup
create  config
create  config/routes.rb
create  config/application.rb
create  config/environment.rb
create  config/secrets.yml
create  config/environments
create  config/environments/development.rb
create  config/environments/production.rb
create  config/environments/test.rb
create  config/initializers
create  config/initializers/assets.rb
create  config/initializers/backtrace_silencers.rb
create  config/initializers/cookies_serializer.rb
create  config/initializers/filter_parameter_logging.rb
create  config/initializers/inflections.rb
```

```
create  config/locales/en.yml
create  config/boot.rb
create  config/database.yml
create  db
create  db/seeds.rb
create  lib
create  lib/tasks
create  lib/tasks/.keep
create  lib/assets
create  lib/assets/.keep
create  log
create  log/.keep
create  public
create  public/404.html
create  public/422.html
create  public/500.html
create  public/favicon.ico
create  public/robots.txt
create  test/fixtures
create  test/fixtures/.keep
create  test/controllers
create  test/controllers/.keep
create  test/mailers
create  test/mailers/.keep
create  test/models
create  test/models/.keep
create  test/helpers
create  test/helpers/.keep
create  test/integration
create  test/integration/.keep
create  test/test_helper.rb
create  tmp/cache
create  tmp/cache/assets
create  vendor/assets/javascripts
create  vendor/assets/javascripts/.keep
create  vendor/assets/stylesheets
create  vendor/assets/stylesheets/.keep
run  bundle install
```

Crearea unei aplicatii rails

```
n@grazvan-HP:~/ruby/web_tech$ rails g scaffold MyTable user_name:string euros:float euros_int:integer
invoke  active_record
create  db/migrate/20150511150740_create_my_tables.rb
create  app/models/my_table.rb
invoke  test_unit
create  test/models/my_table_test.rb
create  test/fixtures/my_tables.yml
invoke  resource_route
route   resources :my_tables
invoke  scaffold_controller
create  app/controllers/my_tables_controller.rb
invoke  erb
create  app/views/my_tables
create  app/views/my_tables/index.html.erb
create  app/views/my_tables/edit.html.erb
create  app/views/my_tables/show.html.erb
create  app/views/my_tables/new.html.erb
create  app/views/my_tables/_form.html.erb
invoke  test_unit
create  test/controllers/my_tables_controller_test.rb
invoke  helper
create  app/helpers/my_tables_helper.rb
invoke  test_unit
invoke  jbuilder
create  app/views/my_tables/index.json.jbuilder
create  app/views/my_tables/show.json.jbuilder
invoke  assets
invoke  coffee
create  app/assets/javascripts/my_tables.coffee
invoke  scss
create  app/assets/stylesheets/my_tables.scss
invoke  scss
create  app/assets/stylesheets/scaffolds.scss
```

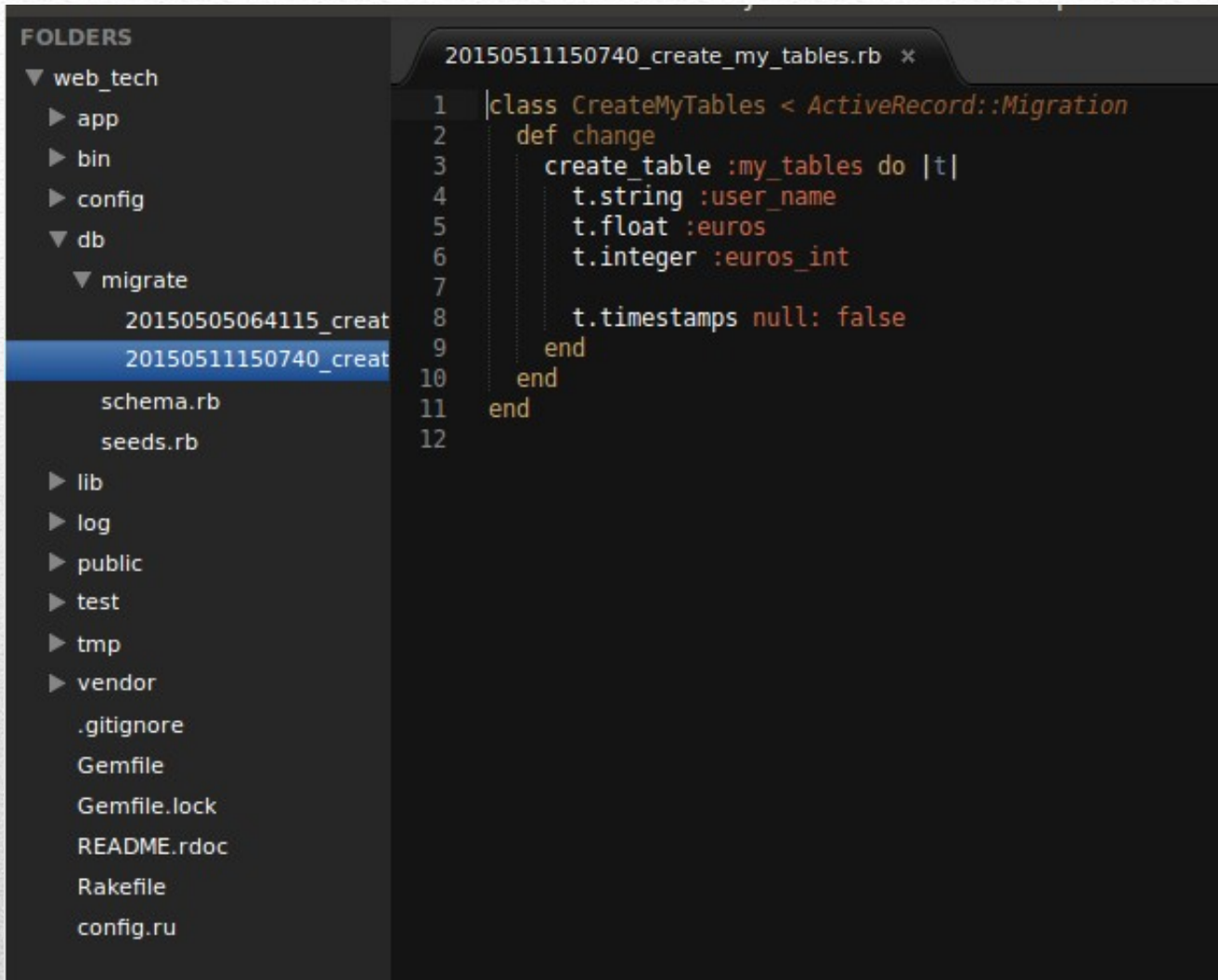
Generare resurse

```
razvan@razvan-HP:~/ruby/web_tech$ rails g model Sondaj id_s:integer id_intrb:integer var_r:string details:text
razvan@razvan-HP:~/ruby/web_tech$ rake db:migrate
== 20150505064115 CreateSondajs: migrating =====
-- create_table(:sondajs)
   -> 0.1583s
== 20150505064115 CreateSondajs: migrated (0.1584s) =====

== 20150511150740 CreateMyTables: migrating =====
-- create_table(:my_tables)
   -> 0.0892s
== 20150511150740 CreateMyTables: migrated (0.0893s) =====

razvan@razvan-HP:~/ruby/web_tech$
```

Generare structura baza de date

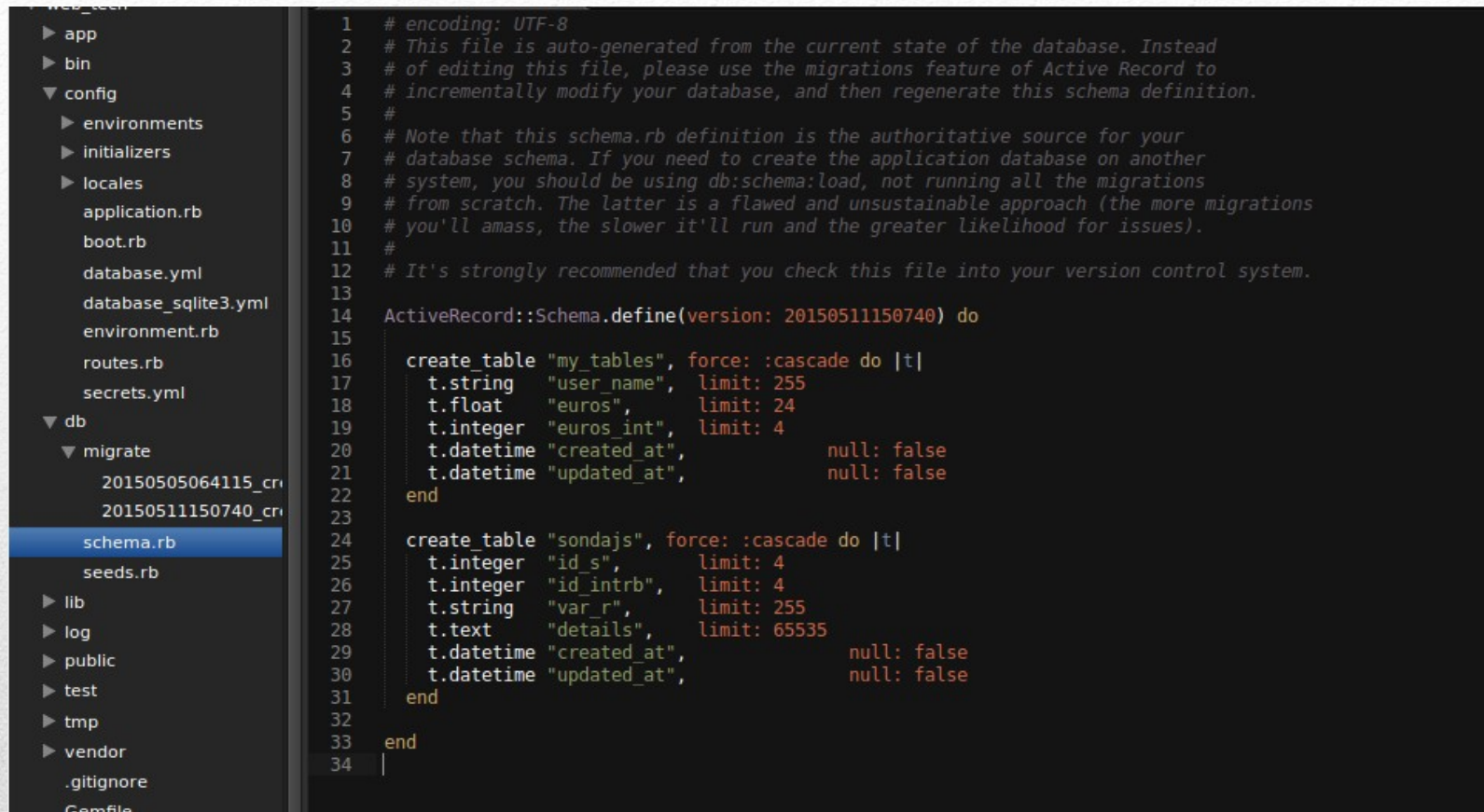


FOLDERS

- ▼ web_tech
 - ▶ app
 - ▶ bin
 - ▶ config
- ▼ db
 - ▼ migrate
 - 20150505064115_creat
 - 20150511150740_creat**
 - schema.rb
 - seeds.rb
- ▶ lib
- ▶ log
- ▶ public
- ▶ test
- ▶ tmp
- ▶ vendor
- .gitignore
- Gemfile
- Gemfile.lock
- README.rdoc
- Rakefile
- config.ru

20150511150740_create_my_tables.rb x

```
1 class CreateMyTables < ActiveRecord::Migration
2   def change
3     create_table :my_tables do |t|
4       t.string :user_name
5       t.float :euros
6       t.integer :euros_int
7
8       t.timestamps null: false
9     end
10  end
11 end
12
```



The image shows a code editor interface. On the left is a file explorer with a tree view. The 'db' folder is expanded, showing a 'migrate' subfolder. Inside 'migrate', there are two files: '20150505064115_create_my_tables.rb' and '20150511150740_create_sondajs.rb'. The '20150511150740_create_sondajs.rb' file is selected and highlighted in blue. The main editor area on the right displays the content of this file, which is a Ruby script for defining a database schema using ActiveRecord. The script includes comments explaining its purpose and the structure of two tables: 'my_tables' and 'sondajs'.

```
1 # encoding: UTF-8
2 # This file is auto-generated from the current state of the database. Instead
3 # of editing this file, please use the migrations feature of Active Record to
4 # incrementally modify your database, and then regenerate this schema definition.
5 #
6 # Note that this schema.rb definition is the authoritative source for your
7 # database schema. If you need to create the application database on another
8 # system, you should be using db:schema:load, not running all the migrations
9 # from scratch. The latter is a flawed and unsustainable approach (the more migrations
10 # you'll amass, the slower it'll run and the greater likelihood for issues).
11 #
12 # It's strongly recommended that you check this file into your version control system.
13
14 ActiveRecord::Schema.define(version: 20150511150740) do
15
16   create_table "my_tables", force: :cascade do |t|
17     t.string   "user_name",   limit: 255
18     t.float    "euros",       limit: 24
19     t.integer  "euros_int",   limit: 4
20     t.datetime "created_at",   null: false
21     t.datetime "updated_at",   null: false
22   end
23
24   create_table "sondajs", force: :cascade do |t|
25     t.integer  "id_s",         limit: 4
26     t.integer  "id_intrb",     limit: 4
27     t.string   "var_r",        limit: 255
28     t.text     "details",      limit: 65535
29     t.datetime "created_at",   null: false
30     t.datetime "updated_at",   null: false
31   end
32
33 end
34
```

Fisierul schema.rb contine structura bazei de date

```
razvan@razvan-HP:~/ruby/web_tech$ rspec
No examples found.

Finished in 0.00027 seconds (files took 0.10299 seconds to load)
0 examples, 0 failures

razvan@razvan-HP:~/ruby/web_tech$ rails g controller html_elem
  create  app/controllers/html_elem_controller.rb
  invoke  erb
  create  app/views/html_elem
  invoke  test_unit
  create  test/controllers/html_elem_controller_test.rb
  invoke  helper
  create  app/helpers/html_elem_helper.rb
  invoke  test_unit
  invoke  assets
  invoke  coffee
  create  app/assets/javascripts/html_elem.coffee
  invoke  scss
  create  app/assets/stylesheets/html_elem.scss
razvan@razvan-HP:~/ruby/web_tech$
```

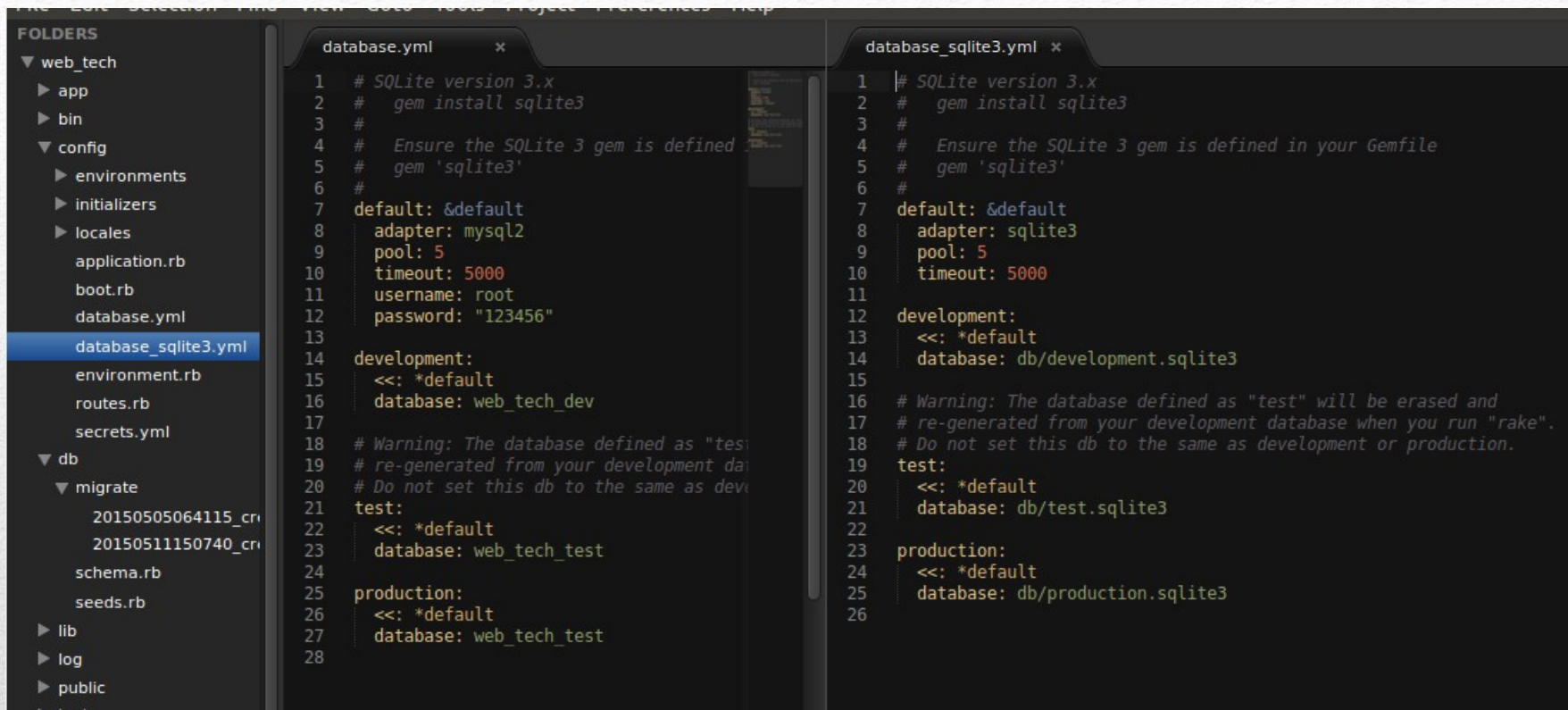
Rularea testelor cu rspec

Si

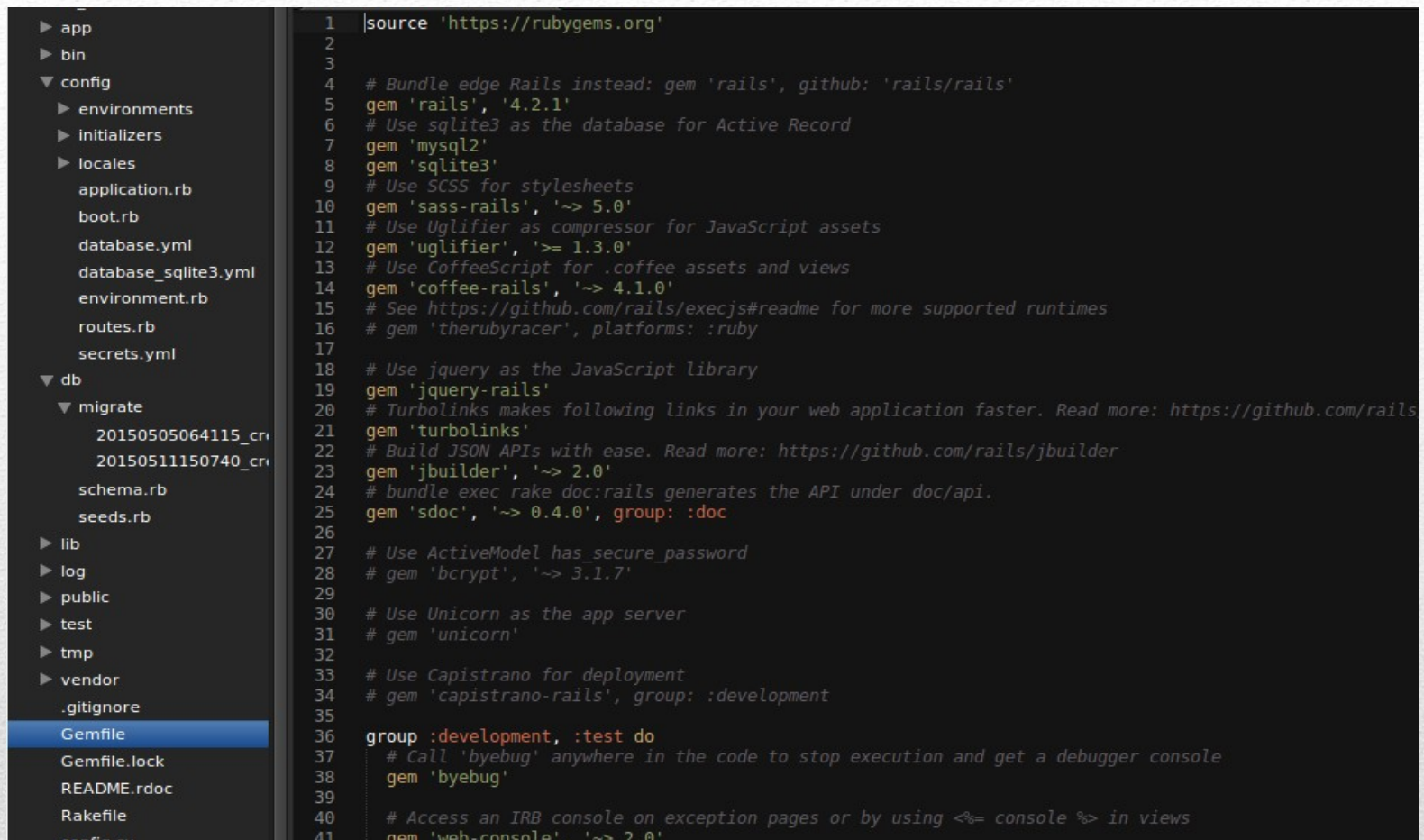
Crearea unui controller

```
1 Rails.application.routes.draw do
2
3   resources :my_tables
4   resources :html_elem, only: :none do
5     collection do
6       get :pers
7       get :hobby
8       get :start
9     end
10  end
11
12   resources :js_elem, only: :none do
13     collection do
14       get :tabel
15     end
16  end
17
18   resources :php_elem, only: :none do
19     collection do
20       get :val_rand
21       post :val_rand_post
22     end
23  end
24
25  # The priority is based upon order of creation: first created gets highest priority
```

Fisierul routes.rb



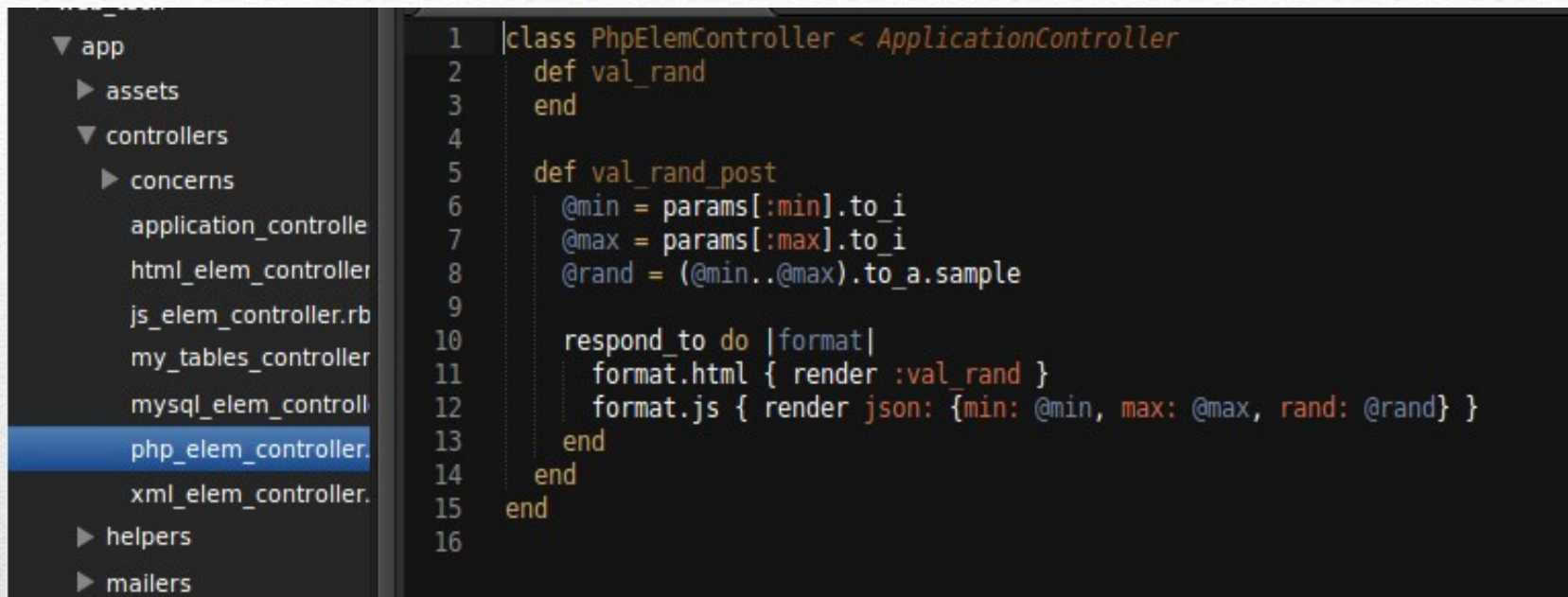
Fisierul database.yml



The image shows a code editor with a sidebar on the left and a main code area on the right. The sidebar displays a file tree with folders like 'app', 'bin', 'config', 'db', 'lib', 'log', 'public', 'test', 'tmp', 'vendor', and files like 'Gemfile', 'Gemfile.lock', 'README.rdoc', 'Rakefile', and 'config.ru'. The 'Gemfile' is selected. The main area shows the content of the Gemfile, which is a Ruby script for managing gems in a Rails application. It includes comments and gem declarations for various dependencies like Rails, database adapters, asset processors, and deployment tools.

```
1 source 'https://rubygems.org'
2
3
4 # Bundle edge Rails instead: gem 'rails', github: 'rails/rails'
5 gem 'rails', '4.2.1'
6 # Use sqlite3 as the database for Active Record
7 gem 'mysql2'
8 gem 'sqlite3'
9 # Use SCSS for stylesheets
10 gem 'sass-rails', '~> 5.0'
11 # Use Uglifier as compressor for JavaScript assets
12 gem 'uglifier', '>= 1.3.0'
13 # Use CoffeeScript for .coffee assets and views
14 gem 'coffee-rails', '~> 4.1.0'
15 # See https://github.com/rails/execjs#readme for more supported runtimes
16 # gem 'therubyracer', platforms: :ruby
17
18 # Use jquery as the JavaScript library
19 gem 'jquery-rails'
20 # Turbolinks makes following links in your web application faster. Read more: https://github.com/rails
21 gem 'turbolinks'
22 # Build JSON APIs with ease. Read more: https://github.com/rails/jbuilder
23 gem 'jbuilder', '~> 2.0'
24 # bundle exec rake doc:rails generates the API under doc/api.
25 gem 'sdoc', '~> 0.4.0', group: :doc
26
27 # Use ActiveModel has_secure_password
28 # gem 'bcrypt', '~> 3.1.7'
29
30 # Use Unicorn as the app server
31 # gem 'unicorn'
32
33 # Use Capistrano for deployment
34 # gem 'capistrano-rails', group: :development
35
36 group :development, :test do
37   # Call 'byebug' anywhere in the code to stop execution and get a debugger console
38   gem 'byebug'
39
40   # Access an IRB console on exception pages or by using <%= console %> in views
41   gem 'web-console', '~> 2.0'
```

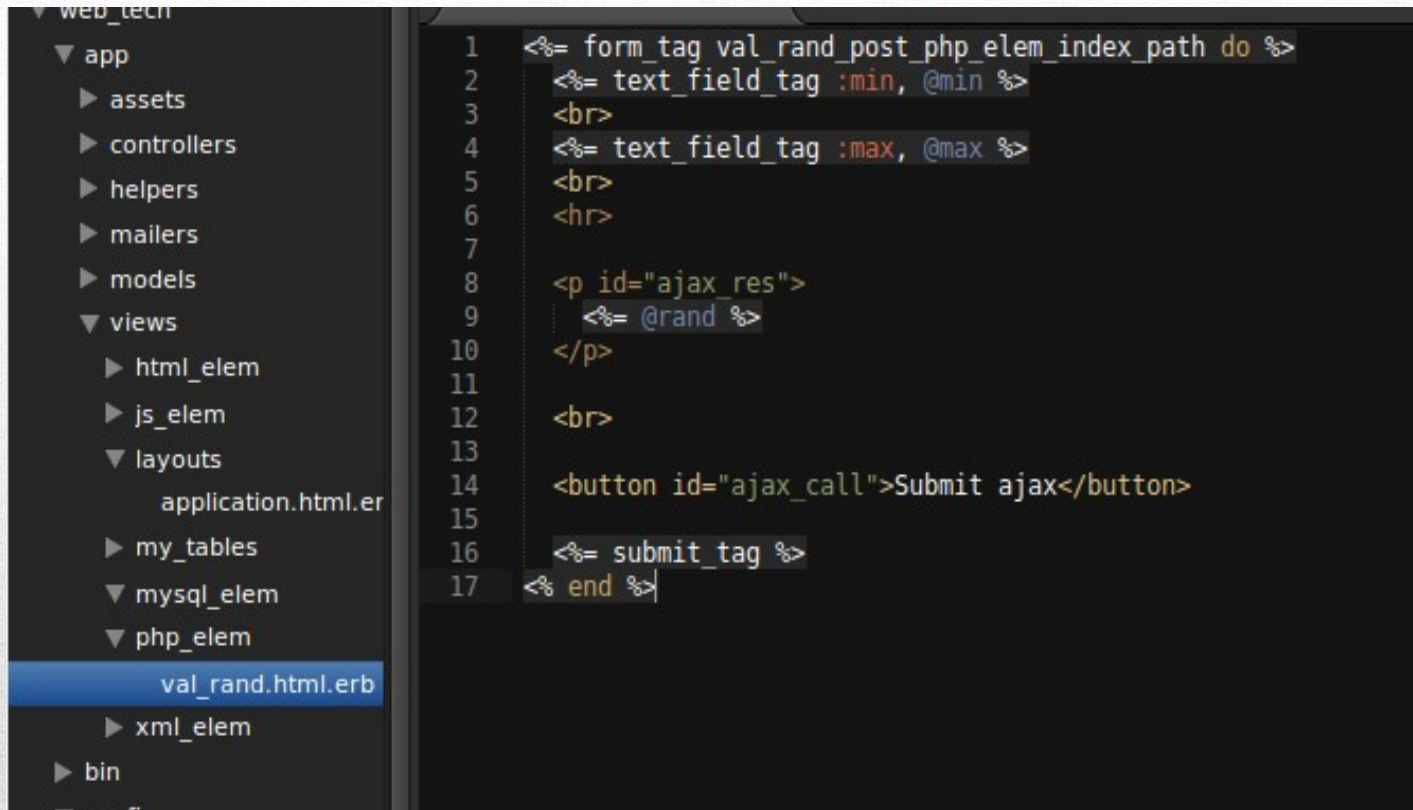
Fisierul Gemfile



The image shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with 'app' expanded, showing 'assets', 'controllers', 'concerns', 'application_controller', 'html_elem_controller', 'js_elem_controller.rb', 'my_tables_controller', 'mysql_elem_controll', 'php_elem_controller.' (highlighted), and 'xml_elem_controller.'. The code editor shows the following code:

```
1 class PhpElemController < ApplicationController
2   def val_rand
3   end
4
5   def val_rand_post
6     @min = params[:min].to_i
7     @max = params[:max].to_i
8     @rand = (@min..@max).to_a.sample
9
10    respond_to do |format|
11      format.html { render :val_rand }
12      format.js { render json: {min: @min, max: @max, rand: @rand} }
13    end
14  end
15 end
16
```

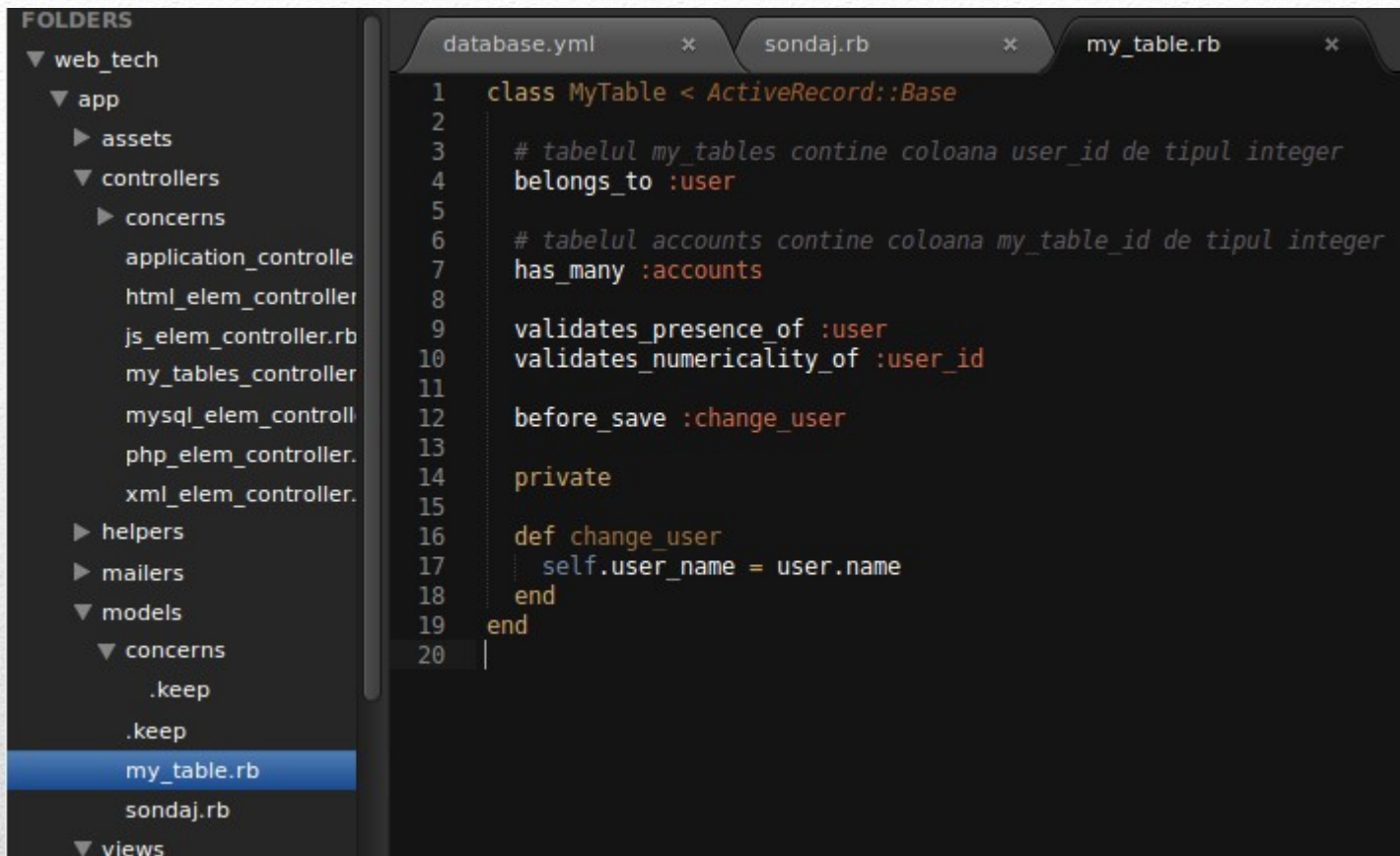
Un controller care returneaza diferite tipuri de date in functie de formatul cerut



The image shows a code editor interface. On the left is a file explorer with a tree structure. The 'views' directory is expanded, showing subdirectories like 'html_elem', 'js_elem', 'layouts', 'my_tables', 'mysql_elem', 'php_elem', and 'val_rand.html.erb' (which is selected). On the right, the content of 'val_rand.html.erb' is displayed, showing an ERB template with lines 1 through 17. The template includes form tags for min and max values, a paragraph with a random value, and a submit button.

```
1 <%= form_tag val_rand_post_php_elem_index_path do %>
2   <%= text_field_tag :min, @min %>
3   <br>
4   <%= text_field_tag :max, @max %>
5   <br>
6   <hr>
7
8   <p id="ajax_res">
9     <%= @rand %>
10  </p>
11
12  <br>
13
14  <button id="ajax_call">Submit ajax</button>
15
16  <%= submit_tag %>
17 <% end %>
```

Un fisier din views care randeaza un form ce contine variabile calculate in controller. Acesta este va fi submis catre urlul mapat de catre rute (comanda “rake routes” ne afiseaza toate maparile la care se adauga terminatia “_path”)



Un model
