

PRACTICA CALIFICADA N2

by @Franklin Espinoza Pari

PARTE 3 RAILS

Crear y aplicar la migración

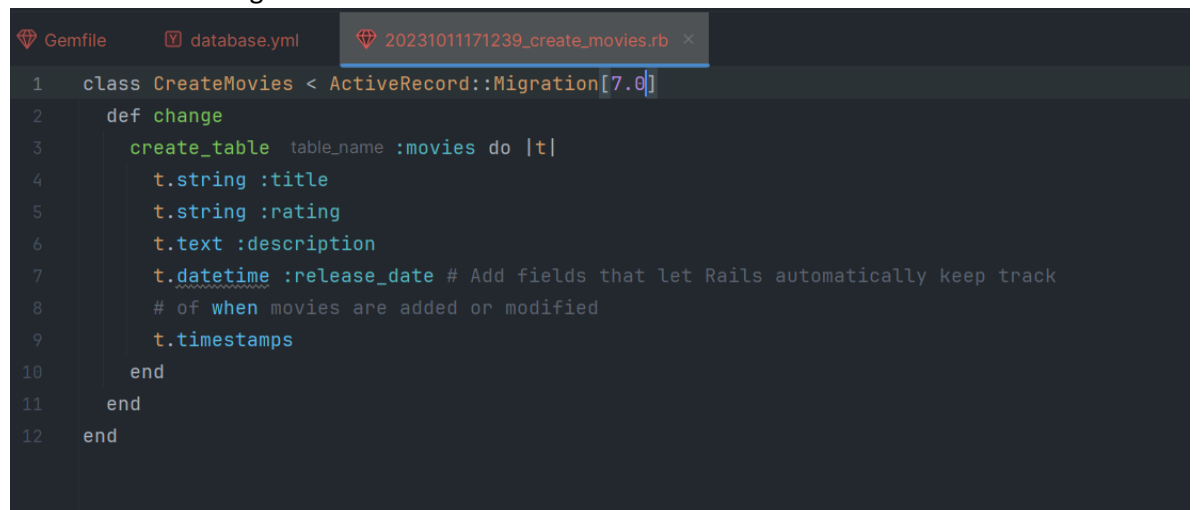
1. Generamos nuestra migración

Ejecutamos el siguiente comando: rails **generate migration create_movies**

```
tion.rb to limit the frameworks that will be loaded.  
PS E:\Sexto Ciclo\CC3S2-Software\Practica Calificada N2\PARTE_3\rottenpotatoes> rails generate migration create_movies  
invoke active_record  
create db/migrate/20231011171239_create_movies.rb  
PS E:\Sexto Ciclo\CC3S2-Software\Practica Calificada N2\PARTE_3\rottenpotatoes>
```

Explicación: El anterior comando genera un nuevo archivo de migración en el directorio db/migrate. Este archivo tiene el siguiente formato “timestamp_create_movie.rb”, donde “timestamp” es la marca de tiempo del momento en que se generó la migración.

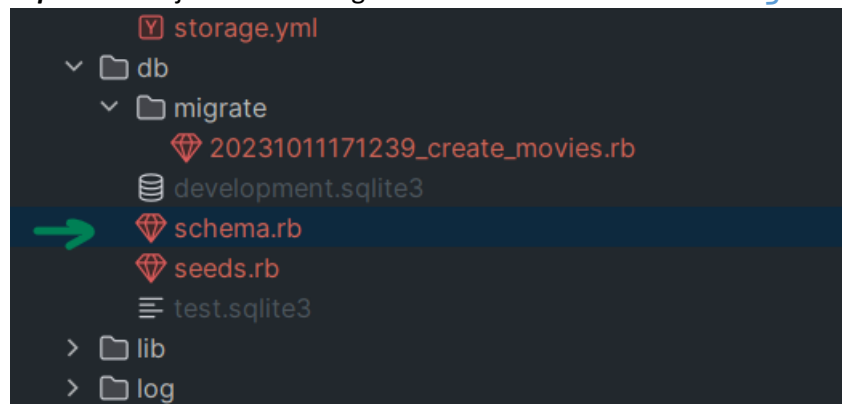
2. Editamos el archivo generado



```
1 class CreateMovies < ActiveRecord::Migration[7.0]  
2   def change  
3     create_table :movies do |t|  
4       t.string :title  
5       t.string :rating  
6       t.text :description  
7       t.datetime :release_date # Add fields that let Rails automatically keep track  
8       # of when movies are added or modified  
9       t.timestamps  
10    end  
11  end  
12 end
```

3. Aplicamos la migración a la base datos de desarrollo

Explicación: Ejecutamos el siguiente comando: rails **db:migrate**



```
storage.yml  
db  
  migrate  
    20231011171239_create_movies.rb  
    development.sqlite3  
    → schema.rb  
    seeds.rb  
    test.sqlite3  
  lib  
  log
```

Una vez ejecutado, se creará el archivo schema.rb

4. Actualizamos el esquema de la base de datos "schema.rb"
Ejecutamos el siguiente comando: `rake db:test:prepare`

Crear el modelo inicial e inicializa la base de datos

1. Crear el modelo Movie

Explicación: En el siguiente directorio app/models crearemos un nuevo archivo movie.rb, el cual será nuestro modelo que hereda de ActiveRecord::Base



```
1 class Movie < ActiveRecord::Base
2 end
3
```

2. Creamos una instancia de Movie



```
PS E:\Sexto Ciclo\CC3S2-Software\Practica Calificada N2\PARTE_3\rottenpotatoes> Rails console
Loading development environment (Rails 7.0.8)
irb(main):001> Movie.new
=> #<Movie:0x00000228b439e700 id: nil, title: nil, rating: nil, description: nil, release_date: nil, created_at: nil, updated_at: nil>
irb(main):002> Movie.first
Movie Load (0.3ms)  SELECT "movies".* FROM "movies" ORDER BY "movies"."id" ASC LIMIT ?  [["LIMIT", 1]]
=> nil
irb(main):003>
```

Explicación: En los resultados anteriores, se ejecuta `Rails console`. Luego, creamos una instancia de Movie con todos sus atributos en nil. Observamos que `Movie.first`, el primer registro de la tabla es igual a `nil` en este caso.

3. Sembrar la base de datos

Primero rellenamos, para eso escribimos el siguiente código en el archivo seeds.rb el cual se encuentra en el directorio db/.



```
9 # Seed the RottenPotatoes DB with some movies.
10 more_movies = [
11   {:title => 'Ganibal', :rating => 'G',
12    :release_date => '25-Nov-1992'},
13   {:title => 'Fuerza bruta', :rating => 'R',
14    :release_date => '21-Jul-1989'},
15   {:title => 'The Ring', :rating => 'PG-13',
16    :release_date => '10-Aug-2011'},
17   {:title => 'Alien: The Return ', :rating => 'PG',
18    :release_date => '12-Jun-1981'}
19 ]
20
21 more_movies.each do | Hash(Symbol->String) movie| Movie.create!(movie) end
22
23
```

Explicación: El archivo seeds.rb se utiliza para poder cargar datos iniciales en la base de datos. Finalmente, ejecutamos `rake db:seed`. Ejecutamos Rails console y verificamos el primer registro de la tabla ejecutando `Movie.first`. Observamos que ahora el primer registro no está vacío como pasaba en el anterior.

```
irb(main):001> Movie.first
  Movie Load (0.1ms)  SELECT "movies".* FROM "movies" ORDER BY "movies"."id" ASC LIMIT ?  [["LIMIT", 1]]
=>
#<Movie:0x0000017fa74bb2e8
 id: 1,
 title: "Ganibal",
 rating: "6",
 description: nil,
 release_date: Wed, 25 Nov 1992 00:00:00.000000000 UTC +00:00,
 created_at: Wed, 11 Oct 2023 18:25:11.920291000 UTC +00:00,
 updated_at: Wed, 11 Oct 2023 18:25:11.920291000 UTC +00:00>
irb(main):002>
```

Adicionales: Usando el método `pluck` podemos obtener en un array los valores de una columna específica. En este caso usamos para la columna `title`.

```
PS E:\Sexto Ciclo\CC3S2-Software\Practica Calificada N2\PARTE_3\rottenpotatoes> Rails console
Loading development environment (Rails 7.0.8)
irb(main):001> Movie.pluck(:title)
irb(main):002> )
  Movie Pluck (0.1ms)  SELECT "movies"."title" FROM "movies"
=> ["Ganibal", "Fuerza bruta", "The Ring", "Alien: The Return "]
irb(main):003>
```

PARTE 4: Crear rutas, acciones y vistas CRUD para películas

1. Visitamos algunos recursos

Explicación: Visitaremos algunos recursos de nuestra aplicación, sin embargo, estas rutas aun no han sido generadas por lo que nos botará un error. Luego verificaremos estos errores en el archivo `log/development.log`, el cual registra los mensajes de error.

Ahora verificamos el error en el archivo development.log

```
Gemfile database.yml 20231011171239_create_movies.rb movie.rb seeds.rb
86 Completed 200 OK in 187ms (Views: 57.1ms | ActiveRecord: 0.0ms | Allocations: 7632)
87
88
89 Started GET "/movie" for 127.0.0.1 at 2023-10-11 13:45:11 -0500
90
91 ActionController::RoutingError (No route matches [GET] "/movie"):
92
93 Started GET "/movies" for 127.0.0.1 at 2023-10-11 13:45:16 -0500
94
95 ActionController::RoutingError (No route matches [GET] "/movies"):
96
97
```

2. Creamos las rutas RESTful
Editamos el archivo config/routes.rb

```
database.yml 20231011171239_create_movies.rb movie.rb seeds.rb development.log routes.rb
1 Rails.application.routes.draw do
2   # Define your application routes per the DSL in https://guides.rubyonrails.org/routing.html
3   # Defines the root path route ("/")
4   # root "articles#index"
5   resources :movies
6   root :to => redirect('/movies')
7 end
8
```

Explicación: En este archivo definimos las rutas para la aplicación. La línea `resources :movies` crea rutas RESTful para el modelo `Movie`. Donde estas rutas incluyen acciones o métodos como `index`, `show`, `new`, `create`, etc.

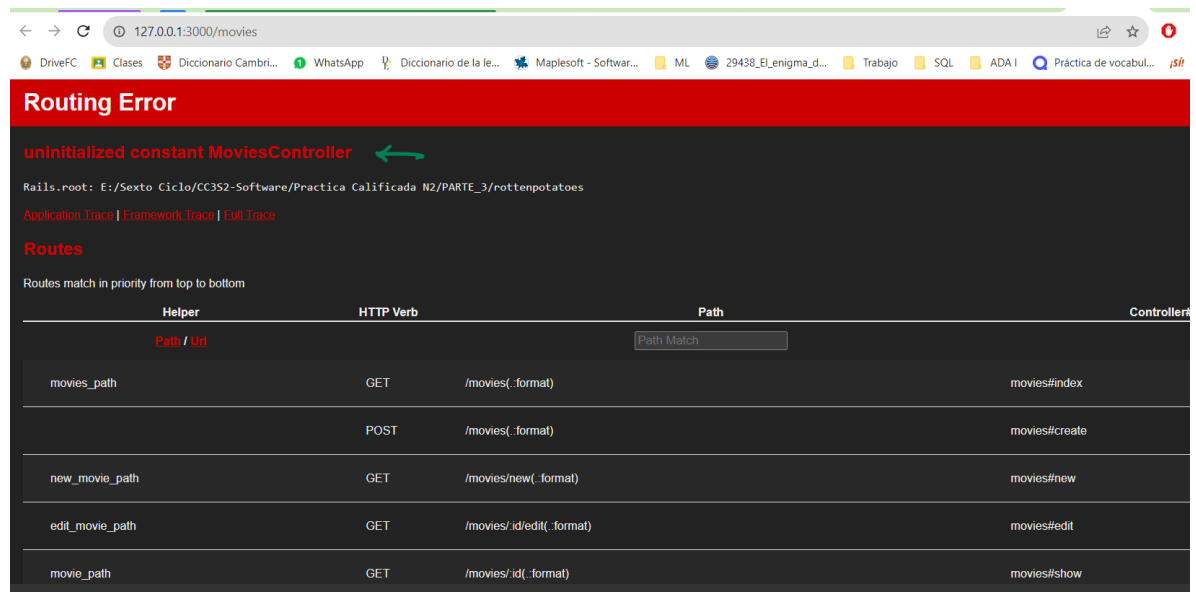
La línea `root :to => redirect('/movies')` lo que hace es establecer la ruta raíz (`/`), y redirige cualquier solicitud a la raíz de la ruta `/movies`, que esta asociada a las rutas RESTful de `'movies'`.

Ejecutamos `rails routes`

```
PS E:\Sexto Ciclo\CC352-Software\Practica Calificada N2\PARTE_3\rottenpotatoes> rails routes

#Action      Prefix Verb  URI Pattern      Controller
ex           movies GET   /movies(.:format) movies#index
ate          movies POST  /movies(.:format) movies#create
ate          new_movie GET   /movies/new(.:format) movies#new
t            edit_movie GET   /movies/:id/edit(.:format) movies#edit
w            movie GET   /movies/:id(.:format) movies#show
ate          PATCH /movies/:id(.:format) movies#update
ate          PUT /movies/:id(.:format) movies#update
ate          DELETE /movies/:id(.:format) movies#destroy
troy         root GET   /                redirect(301, /movies)
0l, /movies) turbo_recede_historical_location GET /recede_historical_location(.:format) turbo/navigation#recede
ve/navigation#recede turbo_resume_historical_location GET /resume_historical_location(.:format) turbo/navigation#resume
ve/navigation#resume turbo_refresh_historical_location GET /refresh_historical_location(.:format) turbo/navigation#refresh
ve/navigation#refresh rails_postmark_inbound_emails POST /rails/action_mailbox/postmark/inbound_emails(.:format) action_mailbox/ingresses/postmark/inbound_emails#create
lbox/ingresses/postmark/inbound_emails#create rails_relay_inbound_emails POST /rails/action_mailbox/relay/inbound_emails(.:format) action_mailbox/ingresses/relay/inbound_emails#create
rails_sendgrid_inbound_emails POST /rails/action_mailbox/sendgrid/inbound_emails(.:format) action_mailbox/ingresses/sendgrid/inbound_emails#create
rails_mandrill_inbound_health_check GET /rails/action_mailbox/mandrill/inbound_emails(.:format) action_mailbox/ingresses/mandrill/inbound_emails#health_check
rails_mandrill_inbound_emails POST /rails/action_mailbox/mandrill/inbound_emails(.:format) action_mailbox/ingresses/mandrill/inbound_emails#create
rails_mailgun_inbound_emails POST /rails/action_mailbox/mailgun/inbound_emails/mime(.:format) action_mailbox/ingresses/mailgun/inbound_emails#create
rails_conductor_inbound_emails GET /rails/conductor/action_mailbox/inbound_emails(.:format)
```

Finalmente, volvemos a correr el servidor y nos ubicamos en el recurso de /movies

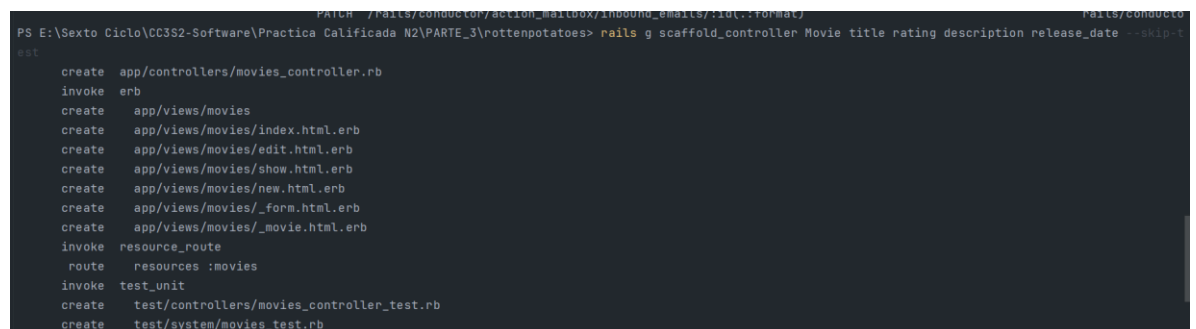


Ahora observamos otro error “uninitialized constant MoviesController”, este error indica que Rails esta tratando de encontrar un controlador llamado “MoviesController” pero no lo encuentra.

3. Crear acciones y vistas

Explicación: El siguiente comando crea un controlador para el modelo Movie con acciones predeterminadas y vistas asociadas. Los campos title, rating, description y release_date se incluirán en las vistas generadas.

```
rails g scaffold_controller Movie title rating description  
release_date --skip-test
```



4. Cambiar la base de datos

Explicación: Para implementar en Heroku, necesitamos asegurarnos de que estamos usando la base de datos PostgreSQL y no SQLite en producción. Esto se hace editando el archivo Gemfile.

```

1 group :production do
2   gem 'pg', '~> 0.21' # Gema para PostgreSQL en producción
3   gem 'rails_12factor' # Mejoras específicas para Heroku
4 end
5
6 group :development, :test do
7   # See https://guides.rubyonrails.org/debugging_rails_applications.html#debug
8   gem "debug", platforms: %i[ mri mingw x64_mingw ]
9   gem 'sqlite3', '~> 1.4.0' # SQLite para entornos de desarrollo y prueba
10 end

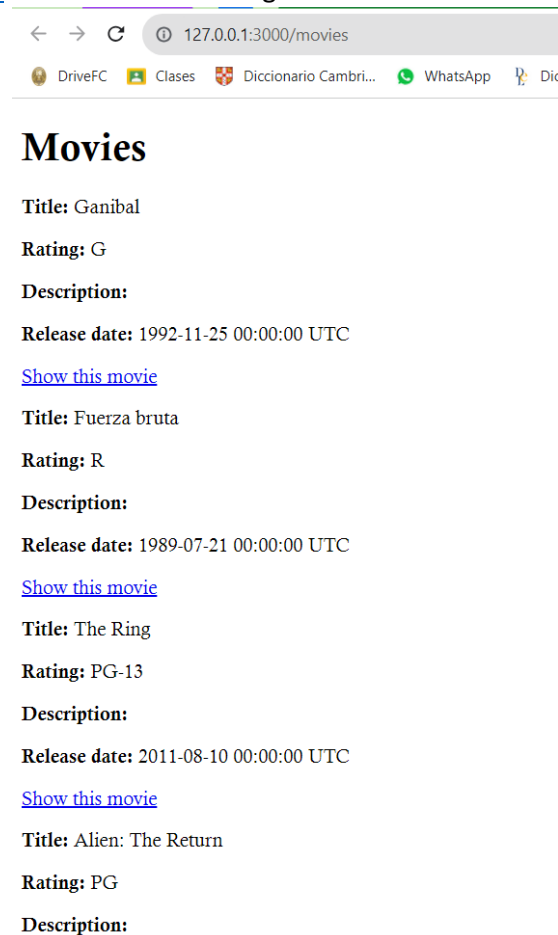
```

Explicación: Esto indica que en producción (:production), uses la gema pg para PostgreSQL y rails_12factor para las mejoras específicas de Heroku. En entornos de desarrollo y prueba (:development, :test), se utiliza SQLite.

Ejecutar `bundle install` o `bundle install --without production` para asegurarse de que las gemas especificadas se instalen correctamente y se actualicen en el Gemfile.lock.

5. Probando en el localhost

Ejecutamos el comando rails server y luego vamos a nuestro navegador y escribimos <http://127.0.0.1:3000>. Se nos mostrara lo siguiente:



← → ↻ 127.0.0.1:3000/movies

DriveFC Clases Diccionario Cambri... WhatsApp Dic

Movies

Title: Ganíbal

Rating: G

Description:

Release date: 1992-11-25 00:00:00 UTC

[Show this movie](#)

Title: Fuerza bruta

Rating: R

Description:

Release date: 1989-07-21 00:00:00 UTC

[Show this movie](#)

Title: The Ring

Rating: PG-13

Description:

Release date: 2011-08-10 00:00:00 UTC

[Show this movie](#)

Title: Alien: The Return

Rating: PG

Description:

6. Implementación a Heroku

a. Iniciamos sesión en Heroku desde la línea de comandos

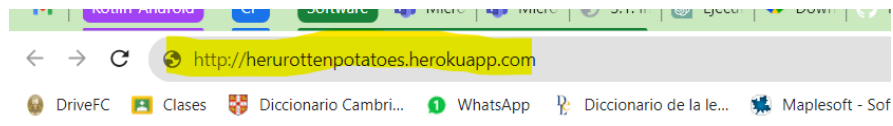
```
PS E:\Sexto Ciclo\CC3S2-Software\Practica Calificada N2\PARTE_3\rottenpotatoes> heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/b88a069a-53e3-48e8-83f8-0139d3711aaa?requestor=WIAAVGA.5MKCY1fQ272g2z6YBTRp3a44vKF5_vf_sCEsx7brYGU
Logging in... done
Logged in as franklin.espinoza0226@gmail.com
PS E:\Sexto Ciclo\CC3S2-Software\Practica Calificada N2\PARTE_3\rottenpotatoes>
```

b. Hacemos el push a heroku

```
PS E:\Sexto Ciclo\CC3S2-Software\Practica Calificada N2\PARTE_3\rottenpotatoes> heroku apps
=== franklin.espinoza0226@gmail.com Apps
herurottenpotatoes
rotten-potatoes

PS E:\Sexto Ciclo\CC3S2-Software\Practica Calificada N2\PARTE_3\rottenpotatoes> heroku git:remote -a herurottenpotatoes
set git remote heroku to https://git.heroku.com/herurottenpotatoes.git
PS E:\Sexto Ciclo\CC3S2-Software\Practica Calificada N2\PARTE_3\rottenpotatoes> git push heroku master
Everything up-to-date
```

c. <http://herurottenpotatoes.herokuapp.com>



Movies

Title: Ganibal

Rating: G

Description:

Release date: 1992-11-25 00:00:00 UTC

[Show this movie](#)

Title: Fuerza bruta

Rating: R

Description:

Release date: 1989-07-21 00:00:00 UTC

[Show this movie](#)

Title: The Ring

Rating: PG-13

Description:

Release date: 2011-08-10 00:00:00 UTC

[Show this movie](#)

Title: Alien: The Return

Rating: PG

Description: