



**Materia: Laboratorio de Sistemas Operativos y
Redes**

Profesor: José Luis Di Biase

Grupo: Lucas Fedele / Victor Hugo Zanardi

A- Introducción

Rails:

Rails es un framework de desarrollo de aplicaciones web escrito en el lenguaje de programación Ruby. Está diseñado para desarrollar aplicaciones web de una manera más fácil y intuitiva, haciendo supuestos sobre lo que cada programador necesita para comenzar. Nos permite escribir menos código obteniendo más resultados que otros lenguajes y frameworks.

Rails es un software dogmático. Éste asume que existe una forma "mejor" de hacer las cosas, y está diseñado para fomentar esa forma - y en algunos casos para desalentar alternativas.

La filosofía de Rails se basa en estos dos principios:

DRY (del inglés, "Don't Repeat Yourself") - sugiere que escribir el mismo código una y otra vez es una mala práctica.

"Convención sobre Configuración" - significa que Rails hace algunas suposiciones sobre lo que quieres hacer y cómo vas a hacerlo, en lugar de requerir que especifiques cada pequeña cosa a través de un sin fin de archivos de configuración.

Cucumber:

Cucumber es una herramienta, escrita en Ruby, que ejecuta descripciones funcionales en texto plano como test automáticos, aprovechando las ventajas del BDD como puede ser acercar la capa de negocio y tecnología de una empresa, permitiendo que no solo se desarrollen bien las funcionalidades, sino que se desarrollen las funcionalidades que más valor van a aportar al usuario, a la vez que definimos casos de prueba y documentamos un proyecto.

La idea principal es que pueda ser entendido y usado no solo por desarrolladores sino por los clientes y gente de negocio.

B- Instalación

1. Procederemos a instalar ruby, rails y cucumber

```
$ sudo apt-get install ruby ruby-dev ruby-rails cucumber
```

En la opinión de los autores de esta guía, el paquete ruby-rails es el más importante de todo el proceso. Ya que existen muchas formas distintas de instalar rails y TODAS menos la forma que te provee tu sistema operativo traen innumerables conflictos con dependencias, gemas, extensiones nativas y librerías de C/C++ sobre las cuales el ruby y sus librerías están implementados.

2. Luego, vamos a instalar un gestor de gemas para encargarnos de las dependencias:

```
$ sudo gem install bundler
```

C- Configuración:

1. Para utilizar rails, crearemos un nuevo proyecto con el siguiente comando:

```
# rails new *nombre del proyecto*
```

esto lo que hará es crear una carpeta con el nombre de proyecto elegido en el directorio actual. Dentro, podremos encontrar todos los archivos que ruby necesita para levantar un sitio web.

2. Entramos en nuestro proyecto:

```
# cd *nombre del proyecto*
```

A continuación, abriremos el archivo Gemfile y buscaremos el siguiente código:

```
group :development, :test do
  # Call 'byebug' anywhere in the code to stop execution and get a debugger console
  gem 'byebug', platform: :mri
end
```

3. Y lo reemplazamos por este:

```
group :development, :test do
  # Call 'byebug' anywhere in the code to stop execution and get a debugger console
  gem 'byebug', platform: :mri

  gem 'cucumber-rails', :require => false
  gem 'database_cleaner'
  gem 'rspec'
end
```

Esto lo que hace es agregar a nuestro proyecto las gemas que permiten DBB cucumber y rspec.

4. Entonces, vamos a proceder a usar el gestor de gemas que instalamos en el proceso de instalación para que se encargue de agregar al proyecto los paquetes necesarios.
IMPORTANTE: NO USAR CON SUDO!!

```
# bundle install
```

5. Una vez que termino, deberemos preparar nuestro proyecto para utilizar cucumber:

```
# rails generate cucumber:install
```

6. Y para correr cucumber:

```
# bundle exec cucumber
```

Agregamos la gema Devise para tener funcionalidad en el proyecto:

1. Abriremos el archivo Gemfile y agregamos la gema Device al final del mismo:

```
gem 'devise'
```

2. Luego necesitamos agrupar Device con nuestra aplicación. Bundle será el encargado de bajar dicha gema de internet:

```
bundle install
```

3. En el siguiente paso necesitamos generar un archivo de configuración para Devise:

```
rails generate devise:install
```

4. Configuramos los archivos de Environments, dentro del archivo buscamos los archivos `development.rb` y `test.rb`, ponemos lo siguiente:

```
config.action_mailer.default_url_options = { host: 'localhost:3000' }
```

5. Luego creamos una Static Pages:

a. Nos situamos en la carpeta del proyecto:

```
cd Documents/*nombre del proyecto*
```

b. Generamos el controller:

```
rails generate controller static_pages
```

c. Modificamos el controller agregando el metodo home en

:

```
def home
end
```

d. Creamos el file "Home" en `app/views/static_pages/home.html.erb` y le agregamos la siguiente línea:

```
<h1>This is the home page</h1>
```

e. Luego creamos la Route en :
`root 'static_pages#home'`

6. Agregamos el mjs flash en :

```
<% flash.each do |key, value| %>
  <div class="alert alert-<%= key %>" alert-dismissible">
    <button type="button" class="close" data-dismiss="alert"
aria-hidden="true">x</button>
    <%= value %>
  </div>
<% end %>
```

7. Configuramos los correos electrónicos en :

```
config.mailer_sender = 'please-change-me-at-config-initializers-devise@example.com'  
(ademas nos aseguramos que esta linea:  
config.secret_key =  
'1788ecf9deaa49646a8cf657d2c00faedd63033f8bfe3e16c7ddc0808502482a11b640c5b0  
a9f1089696477e7f3199a9861a1594422fccfa4222bb5d151b60fc'  
no esté comentada)
```

8. Y en una terminal ejecutamos (esto generará nuevos archivos):

`rails generate devise user`

9. Añadimos el enlace al Master Layout en

```
<%= link_to 'Login', new_user_session_path %>
```

10. Ejecutamos el siguiente comando en la terminal:

`rake db:migrate`

11. Y levantamos el server:

`rails server`

Con esto ya tendremos la gema Devise instalada y configurada:



Sign up

Email

Password

Password confirmation

[Sign in](#)



Sign in

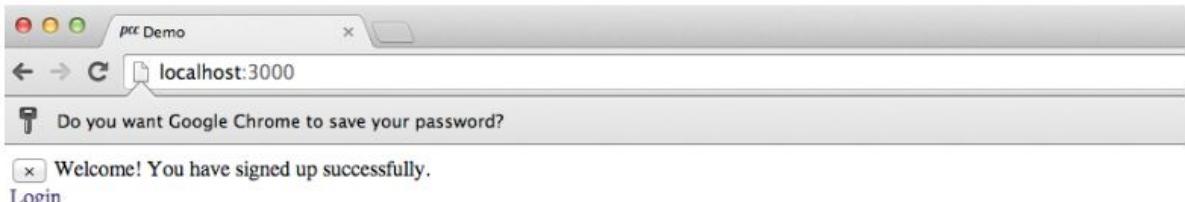
Email

Password

Remember me

[Sign up](#)

[Forgot your password?](#)



4. Problemas

Se intentó seguir el tutorial brindado por la página oficial del framework (<http://www.rubyonrails.org.es/instala.html>) debido a que la instalación por gemas hace que no se instale todas las necesarias para poder desarrollar. Se nos presento el siguiente error:

ERROR: Failed to build gem native extension.

Intentamos solucionar dicho problema usando el comando:

sudo gem update

Pero no tuvimos éxito. Entonces encontramos la segundo variante que es la que presentamos en el tp.