

---

# Tutorial Rápido de Ruby on Rails



Saray Cabrera y David Kang

---

iHola!





Nos presentamos: somos Saray y David y así es nuestro día a día.

Como ven, tenemos una plaga de camaleones... nos pasa a todos cuando empezamos a trabajar en SUSE :-)

# Ruby != Rails



es un lenguaje



es un framework

Ruby y Rails son cosas distintas a pesar de su estrecha relación.

Ruby es un lenguaje de programación. Lenguaje orientado a objetos (todo en ruby es un objeto). Lenguaje interpretado, no compilado.

Ejemplos de lenguajes: Ruby, Java, JavaScript, PHP...

Rails (o Ruby on Rails) es un framework, es decir, una herramienta que nos ayuda a la hora de crear aplicaciones web usando Ruby como lenguaje base.

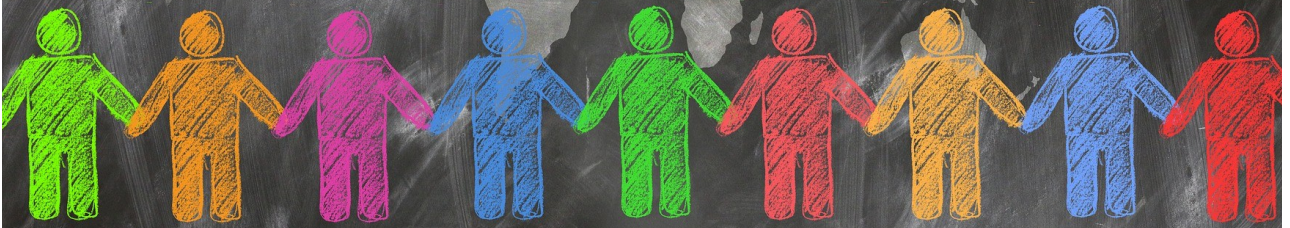
Ejemplos de frameworks: Rails/Ruby on Rails, Laravel para PHP, Django para Python...

¿Por qué Ruby y Rails?

# OPEN SOURCE



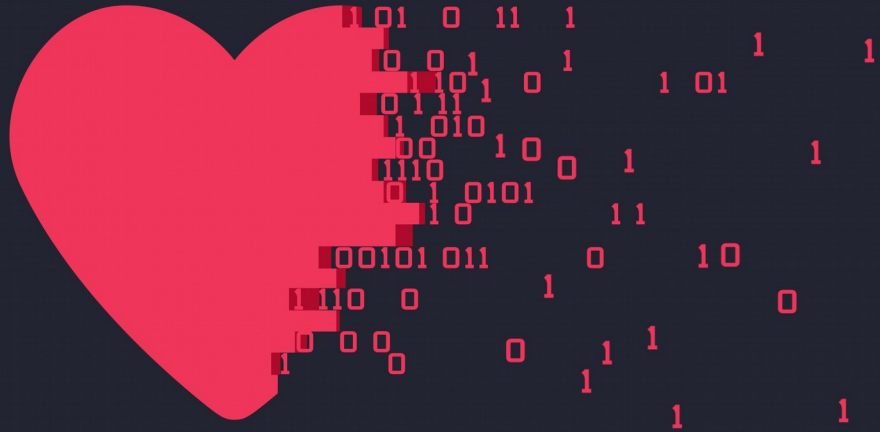
# Comunidad



## Implica:

- buena documentación
- en internet hay muchas soluciones a problemas comunes
- tanto el lenguaje como el framework se mantienen activos: con actualizaciones frecuentes, mejoras constantes, ...

# Nos encanta

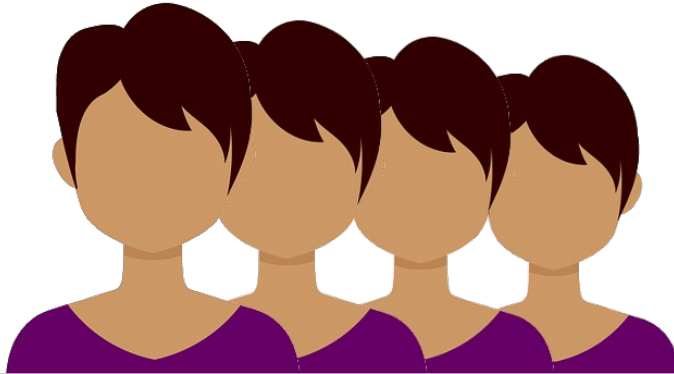


- RUBY
  - Cercano a nuestro lenguaje natural
  - Simple a la hora de leerlo y escribirlo
  - Enfocado en la productividad
  - Nos permite olvidarnos de detalles de la máquina
- RAILS
  - Es el más extendido de los frameworks para Ruby
  - Nos evita quebraderos de cabeza
    - Facilita mucho el desarrollo.
    - Se encarga de las tareas tediosas, haciendo por nosotros tareas/configuraciones frecuentes.
    - Nos permite enfocarnos en la lógica de nuestra aplicación.



# Los 2 Principios de Rails

# DRY (Don't Repeat Yourself)



El principio DRY persigue que no repitamos trabajo o código. Por ejemplo, un elemento que sea útil en varias partes del código se define en un único sitio y se reutiliza en todos los demás. Se consigue simplicidad, consistencia y menor posibilidad de errores, entre otras ventajas.

# Convención sobre Configuración



Siguiendo las convenciones propuestas por Rails ahorramos mucho trabajo de configuración.

Como ejemplo sencillo, supongamos que hemos creado un recurso llamado “Post” en nuestra aplicación. Si seguimos la convención de Rails, sabremos que su correspondiente tabla en la Base de Datos se llamará “posts” y no es necesario indicarlo o configurarlo en ninguna parte. Sólo si quisiéramos que la tabla se llamara de forma distinta (no seguir la convención) habría que especificarlo.

# Otros Frameworks



Sinatra

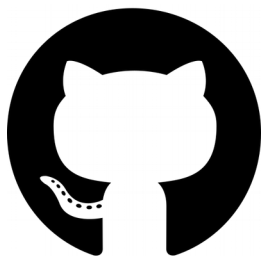


Hanami



Padrino

¿Quién usa Ruby on Rails?



**GROUPON®**

**KICKSTARTER**



# Problemos



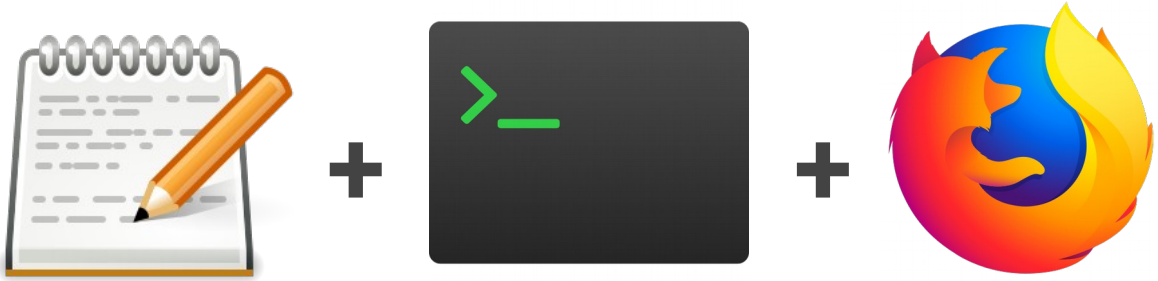
---

<https://ruby.github.io/TryRuby>

---



# ¿Qué necesitamos?



Vim, Emacs, Sublime, Atom, Visual Studio...

---

# Instalación de Ruby y Rails

<http://www.installrails.com>

<https://rvm.io>

---

¡Manos a la obra!

# ¿Qué tenemos en mente?

- Crear el típico Blog sencillo
  - Crear Posts
  - Crear Comentarios
- Darle algo de flow al blog: CSS

# Crear una aplicación

- Terminal
  - rails new <nombre\_app>
  - rails server

Es posible que nos de un error relacionado con ExecJS

Si nos da el error, la forma más rápida y sencilla de solventarlo es abriendo el fichero Gemfile y descomentando la línea que incluye la gema “mini-race”. Ejecutando después “bundle install” para que instale esa nueva gema.

# Gemas

- Son los “Plugins” de Ruby → añade funcionalidad
- Fichero **Gemfile** → gemas que queremos instalar
- Fichero **Gemfile.lock** → gemas instaladas, incluyendo su versión y sus dependencias

<https://rubygems.org>

<https://github.com>

Rails provee un cierto número de gemas que necesita para su funcionamiento o las ofrece por defecto. Pero podemos sustituirlas por otras y, por supuesto, añadir las que queramos.

# Crear una aplicación

- Terminal
  - rails server
- Navegador
  - En nuestro navegador: [localhost:3000](http://localhost:3000)

De esta manera podemos arrancar un servidor en nuestro equipo y podemos acceder a nuestra aplicación a través de "localhost:3000". Así podemos ver el resultado de lo que estamos desarrollando, simulamos de esta manera lo que vería un usuario real desde cualquier parte del mundo si la aplicación estuviera alojada en un servidor web.



Yay! You're on Rails!



**Rails version:** 5.2.3

**Ruby version:** 2.5.5 (x86\_64-linux)

Cuando ejecutamos el primer paso para crear nuestra aplicación y accedemos a localhost:3000, veremos esta página que ofrece Rails por defecto.



# Nuestro primer recurso con el generador

- Terminal
  - rails generate scaffold Post title:string content:text
  - rails db:migrate
- Navegador
  - En nuestro navegador: [localhost:3000/posts](http://localhost:3000/posts)

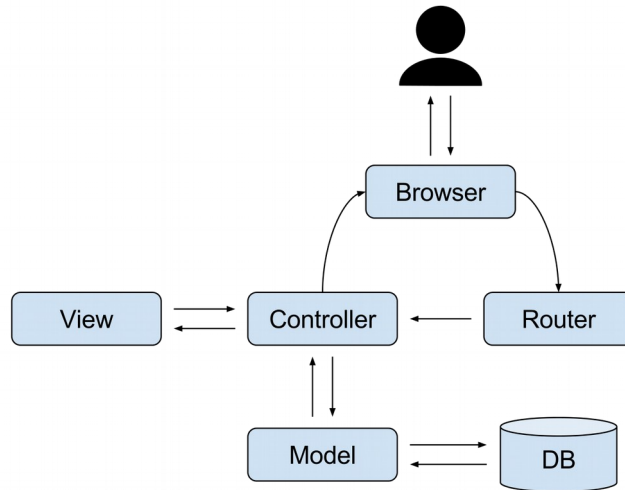
Scaffold (andamio en español), nos facilita la labor de crear nuestros recursos. Con un simple comando crea el fichero de migración para la base de datos, la vista, el modelo, controlador, etc. Útil para dar los primeros pasos con Ruby on Rails o incluso para prototipar, es decir hacer una aplicación extremadamente sencilla para probar o mostrar nuestra idea.

---

# ¿Qué genera el “scaffold”?

- Modelo
  - Vistas
  - Controlador
  - CSS
  - Javascript
  - Tests
  - Helpers
-

# Modelo Vista Controlador (MVC)



Rails sigue el patrón Modelo-Vista-Controlador. Este artículo tiene una muy breve explicación que puede resultar útil:

<https://www.railstutorial.org/book/beginning#sec-mvc>

---

# Otro recurso y su asociación

- Terminal
  - rails generate scaffold Comment content:text post:references
  - rails db:migrate
- Editor
  - Añadimos asociación
    - has\_many :comments #en el modelo de post
- Navegador
  - En nuestro navegador: [localhost:3000/comments](http://localhost:3000/comments)

---

# Modificamos las vistas

- Editor
  - Reemplazamos la referencia al objeto post por su título
  - Añadimos listado de comentarios a la página del post
  - Añadimos un menú

# ¡Podemos hacerlo más bonito!



Modificando los ficheros CSS que encontramos en el directorio `app/assets/stylesheets/*` podemos darle algo de estilo a nuestras páginas. Color, tipografía, espaciado, etc.

# Enlaces Útiles

<https://www.ruby-lang.org>

<https://guides.rubyonrails.org>

<https://ruby.github.io/TryRuby>

<https://rubygems.org>

<http://railsgirls.com>

<https://www.railstutorial.org>

¿Preguntas? ☒



¡GRACIAS! 😊



@sarayacp



@D4vid\_Kang

