

Ruby on Rails

Una breve introducción

Mancuso Emiliano

`www.elBlogolico.com.ar`

2009



Contenidos del curso

Ruby

Que es

Ventajas

Desventajas

Ejemplos

Rails

Que es

Componentes

Aplicación en Rails

Contenidos del curso

Ruby

Que es

Ventajas

Desventajas

Ejemplos

Rails

Que es

Componentes

Aplicación en Rails

Hablemos sobre..

Ruby

Que es

Ventajas

Desventajas

Ejemplos

Rails

Que es

Componentes

Aplicación en Rails



Que es

- ▶ Ruby es un lenguaje de programación
 - ▶ Orientado a objetos
 - ▶ Interpretado
 - ▶ Reflexivo



Que es

- ▶ Ruby es un lenguaje de programación
 - ▶ Orientado a objetos
 - ▶ Interpretado
 - ▶ Reflexivo



Que es

- ▶ Ruby es un lenguaje de programación
 - ▶ Orientado a objetos
 - ▶ Interpretado
 - ▶ Reflexivo



Que es

- ▶ Ruby es un lenguaje de programación
 - ▶ Orientado a objetos
 - ▶ Interpretado
 - ▶ Reflexivo



Hablemos sobre..

Ruby

Que es

Ventajas

Desventajas

Ejemplos

Rails

Que es

Componentes

Aplicación en Rails



Ventajas

- ▶ **Referencia a objetos**

- ▶ Soporte herencia

Ventajas

- ▶ **Referencia a objetos**

Las variables siempre son referencias a objetos, no los objetos mismos

- ▶ Soporte herencia



Ventajas

- ▶ Referencia a objetos
- ▶ **Soporte herencia**

Ventajas

- ▶ Referencia a objetos
- ▶ **Soporte herencia**
 - ▶ Enlace Dinámico
 - ▶ Mixins



Ventajas

- ▶ Referencia a objetos
- ▶ **Soporte herencia**
 - ▶ **Enlace Dinámico**
 - ▶ Librerías enlazadas en ejecución
 - ▶ Más liviano
 - ▶ Evita duplicación de código
 - ▶ Mixins



Ventajas

- ▶ Referencia a objetos
- ▶ **Soporte herencia**
 - ▶ **Enlace Dinámico**
 - ▶ Librerías enlazadas en ejecución
 - ▶ Más liviano
 - ▶ Evita duplicación de código
 - ▶ Mixins



Ventajas

- ▶ Referencia a objetos
- ▶ **Soporte herencia**
 - ▶ **Enlace Dinámico**
 - ▶ Librerías enlazadas en ejecución
 - ▶ Más liviano
 - ▶ Evita duplicación de código
 - ▶ Mixins



Ventajas

- ▶ Referencia a objetos
- ▶ **Soporte herencia**
 - ▶ Enlace Dinámico
 - ▶ **Mixins**

Como las interfaces, pero con su implementación



Ventajas

- ▶ Referencia a objetos
- ▶ Soporte herencia
 - ▶ Enlace Dinámico
 - ▶ Mixins

Otras ventajas

- ▶ **Metaprogramación**
- ▶ Hilos
- ▶ Tipificado dinámico
- ▶ Expresiones regulares
- ▶ Garbage collector automático
- ▶ Sintaxis simple



Otras ventajas

- ▶ **Metaprogramación**
- ▶ **Hilos**
- ▶ Tipificado dinámico
- ▶ Expresiones regulares
- ▶ Garbage collector automático
- ▶ Sintaxis simple



Otras ventajas

- ▶ Metaprogramación
- ▶ Hilos
- ▶ Tipificado dinámico
- ▶ Expresiones regulares
- ▶ Garbage collector automático
- ▶ Sintaxis simple



Otras ventajas

- ▶ Metaprogramación
- ▶ Hilos
- ▶ Tipificado dinámico
- ▶ Expresiones regulares
- ▶ Garbage collector automático
- ▶ Sintaxis simple



Otras ventajas

- ▶ Metaprogramación
- ▶ Hilos
- ▶ Tipificado dinámico
- ▶ Expresiones regulares
- ▶ Garbage collector automático
- ▶ Sintaxis simple



Otras ventajas

- ▶ Metaprogramación
- ▶ Hilos
- ▶ Tipificado dinámico
- ▶ Expresiones regulares
- ▶ Garbage collector automático
- ▶ Sintaxis simple



Hablemos sobre..

Ruby

Que es

Ventajas

Desventajas

Ejemplos

Rails

Que es

Componentes

Aplicación en Rails



Desventajas

- ▶ No soporta polimorfismo de funciones (sobrecarga)
- ▶ No es tipado fuerte
- ▶ Interpretado = Lento



Desventajas

- ▶ No soporta polimorfismo de funciones (sobrecarga)
- ▶ No es tipado fuerte
- ▶ Interpretado = Lento



Desventajas

- ▶ No soporta polimorfismo de funciones (sobrecarga)
- ▶ No es tipado fuerte
- ▶ Interpretado = Lento



Hablemos sobre..

Ruby

Que es

Ventajas

Desventajas

Ejemplos

Rails

Que es

Componentes

Aplicación en Rails



Ejemplos

Ruby

```
1 puts "Hola Blogolico"
2
3 => "Hola Blogolico"
4
5 def tripleSaludo
6   "Hola Blogolico!\n" * 3
7 end
8
9 puts tripleSaludo
10
11 => "Hola Blogolico!"
12 => "Hola Blogolico!"
13 => "Hola Blogolico!"
```



Hablemos sobre..

Ruby

Que es

Ventajas

Desventajas

Ejemplos

Rails

Que es

Componentes

Aplicación en Rails



Que es Rails

- ▶ Framework de aplicaciones web escrito en Ruby
- ▶ Paradigma MVC
 - ▶ Modelo
 - ▶ Vista
 - ▶ Controlador

Que es Rails

- ▶ Framework de aplicaciones web escrito en Ruby
- ▶ Paradigma MVC
 - ▶ Modelo
 - ▶ Vista
 - ▶ Controlador



Que es Rails

- ▶ Framework de aplicaciones web escrito en Ruby
- ▶ Paradigma MVC
 - ▶ **M**odelo
Clases que representan el modelo en la base de datos
 - ▶ **V**ista
 - ▶ **C**ontrolador



Que es Rails

- ▶ Framework de aplicaciones web escrito en Ruby
- ▶ Paradigma MVC
 - ▶ **M**odelo
 - ▶ **V**ista
Interfaz de usuario. Datos del controlador
 - ▶ **C**ontrolador



Que es Rails

- ▶ Framework de aplicaciones web escrito en Ruby
- ▶ Paradigma MVC
 - ▶ **M**odelo
 - ▶ **V**ista
 - ▶ **C**ontrolador

Eventos, acciones, cambios en el modelo



Que es Rails

- ▶ Framework de aplicaciones web escrito en Ruby
- ▶ Paradigma MVC
 - ▶ **M**odelo
 - ▶ **V**ista
 - ▶ **C**ontrolador



Filosofía

- ▶ **DRY *Don't Repeat Yourself***
- ▶ Convention Over Configuration
- ▶ REST

Es un mal hábito volver a escribir el mismo código

Filosofía

- ▶ *DRY Don't Repeat Yourself*
- ▶ **Convention Over Configuration**
- ▶ REST

*Rails supone que quieres hacer y como lo vas a hacer.
No te preocupes por la configuración.*



Filosofía

- ▶ *DRY Don't Repeat Yourself*
- ▶ *Convention Over Configuration*
- ▶ **REST**

Patrón de diseño para aplicaciones web.



Filosofía

- ▶ *DRY Don't Repeat Yourself*
- ▶ Convention Over Configuration
- ▶ REST

Hablemos sobre..

Ruby

Que es

Ventajas

Desventajas

Ejemplos

Rails

Que es

Componentes

Aplicación en Rails



Para la creación de aplicaciones web

- ▶ Action Controller
- ▶ Action View
- ▶ Active Record
- ▶ Action Mailer
- ▶ Active Resource
- ▶ Railties
- ▶ Active Support



Para la creación de aplicaciones web

- ▶ **Action Controller**
- ▶ Action View
- ▶ Active Record
- ▶ Action Mailer
- ▶ Active Resource
- ▶ Railties
- ▶ Active Support

Maneja los controladores de la aplicación.

Procesa peticiones, extrae parametros, ejecuta.

Ejemplo: gestion de sesiones, template rendering, etc.



Para la creación de aplicaciones web

- ▶ Action Controller
- ▶ Action View
- ▶ Active Record
- ▶ Action Mailer
- ▶ Active Resource
- ▶ Railties
- ▶ Active Support

Maneja las vistas de la aplicación.
Puede crear HTML y XML.

Incluye soporte para AJAX built-in.



Para la creación de aplicaciones web

- ▶ Action Controller
- ▶ Action View
- ▶ Active Record
- ▶ Action Mailer
- ▶ Active Resource
- ▶ Railties
- ▶ Active Support

Base de los modelos de datos.

Provee independencia de la DB, funcionalidad básica CRUD, capacidad avanzada de búsqueda, relaciona modelos, etc.



Para la creación de aplicaciones web

- ▶ Action Controller
- ▶ Action View
- ▶ Active Record
- ▶ Action Mailer
- ▶ Active Resource
- ▶ Railties
- ▶ Active Support

Framework para crear servicios de e-mail.

Se puede usar para enviar, recibir y procesar e-mails.



Para la creación de aplicaciones web

- ▶ Action Controller
- ▶ Action View
- ▶ Active Record
- ▶ Action Mailer
- ▶ Active Resource
- ▶ Railties
- ▶ Active Support

Framework para gestionar la conexión entre objetos de negocio y RESTful web services.

Implementa el mapeo entre estos, con la semántica CRUD.



Para la creación de aplicaciones web

- ▶ Action Controller
- ▶ Action View
- ▶ Active Record
- ▶ Action Mailer
- ▶ Active Resource
- ▶ Railties
- ▶ Active Support

Código del núcleo de Rails que crea nuevas aplicaciones y las conecta con los frameworks en una sola aplicación.



Para la creación de aplicaciones web

- ▶ Action Controller
- ▶ Action View
- ▶ Active Record
- ▶ Action Mailer
- ▶ Active Resource
- ▶ Railties
- ▶ Active Support

Gran colección de clases y extensiones de la biblioteca estandar de Ruby, que usa Rails tanto el núcleo como tus aplicaciones.



Para la creación de aplicaciones web

- ▶ Action Controller
- ▶ Action View
- ▶ Active Record
- ▶ Action Mailer
- ▶ Active Resource
- ▶ Railties
- ▶ Active Support



Hablemos sobre..

Ruby

Que es

Ventajas

Desventajas

Ejemplos

Rails

Que es

Componentes

Aplicación en Rails



Antes de empezar..

Necesitamos tener instalado

- ▶ Ruby
- ▶ RubyGems
 - ▶ Rails *gem install rails*
 - ▶ SQLite *gem install sqlite3-ruby*



Antes de empezar..

Necesitamos tener instalado

- ▶ Ruby
- ▶ RubyGems
 - ▶ Rails *gem install rails*
 - ▶ SQLite *gem install sqlite3-ruby*



Antes de empezar..

Necesitamos tener instalado

- ▶ Ruby
- ▶ RubyGems
 - ▶ Rails *gem install rails*
 - ▶ SQLite *gem install sqlite3-ruby*



Antes de empezar..

Necesitamos tener instalado

- ▶ Ruby
- ▶ RubyGems
 - ▶ Rails *gem install rails*
 - ▶ SQLite *gem install sqlite3-ruby*



Primer aplicación

- ▶ **Creamos el directorio para nuestras aplicaciones**
- ▶ Creamos nuestra aplicacion rails
- ▶ Accedemos al directorio
- ▶ Creamos la base de datos
- ▶ Creamos el controlador para saludar
- ▶ Editamos la vista del controlador
- ▶ Agregamos el codigo del saludo en el index
- ▶ Arrancamos el servidor, junto con la aplicación
- ▶ Abrimos un navegador, con la dirección del servidor

```
mkdir ~/railsApps
```

Primer aplicación

- ▶ Creamos el directorio para nuestras aplicaciones
- ▶ **Creamos nuestra aplicacion rails**
- ▶ Accedemos al directorio
- ▶ Creamos la base de datos
- ▶ Creamos el controlador para saludar
- ▶ Editamos la vista del controlador
- ▶ Agregamos el codigo del saludo en el index
- ▶ Arrancamos el servidor, junto con la aplicación
- ▶ Abrimos un navegador, con la dirección del servidor

```
rails blogolicoApp
```



Primer aplicación

- ▶ Creamos el directorio para nuestras aplicaciones
- ▶ Creamos nuestra aplicacion rails
- ▶ **Accedemos al directorio**
- ▶ Creamos la base de datos
- ▶ Creamos el controlador para saludar
- ▶ Editamos la vista del controlador
- ▶ Agregamos el codigo del saludo en el index
- ▶ Arrancamos el servidor, junto con la aplicación
- ▶ Abrimos un navegador, con la dirección del servidor

```
cd ~/railsApps/blogolicoApp
```



Primer aplicación

- ▶ Creamos el directorio para nuestras aplicaciones
- ▶ Creamos nuestra aplicacion rails
- ▶ Accedemos al directorio
- ▶ **Creamos la base de datos**
- ▶ Creamos el controlador para saludar
- ▶ Editamos la vista del controlador
- ▶ Agregamos el codigo del saludo en el index
- ▶ Arrancamos el servidor, junto con la aplicación
- ▶ Abrimos un navegador, con la dirección del servidor

```
rake db:create
```



Primer aplicación

- ▶ Creamos el directorio para nuestras aplicaciones
- ▶ Creamos nuestra aplicacion rails
- ▶ Accedemos al directorio
- ▶ Creamos la base de datos
- ▶ **Creamos el controlador para saludar**
 - ▶ Editamos la vista del controlador
 - ▶ Agregamos el codigo del saludo en el index
 - ▶ Arrancamos el servidor, junto con la aplicación
 - ▶ Abrimos un navegador, con la dirección del servidor

```
ruby script/generate controller saludo index
```



Primer aplicación

- ▶ Creamos el directorio para nuestras aplicaciones
- ▶ Creamos nuestra aplicacion rails
- ▶ Accedemos al directorio
- ▶ Creamos la base de datos
- ▶ Creamos el controlador para saludar
- ▶ **Editamos la vista del controlador**
 - ▶ Agregamos el codigo del saludo en el index
 - ▶ Arrancamos el servidor, junto con la aplicación
 - ▶ Abrimos un navegador, con la dirección del servidor

```
vi ./app/views/saludo/index.html.erb
```



Primer aplicación

- ▶ Creamos el directorio para nuestras aplicaciones
- ▶ Creamos nuestra aplicacion rails
- ▶ Accedemos al directorio
- ▶ Creamos la base de datos
- ▶ Creamos el controlador para saludar
- ▶ Editamos la vista del controlador
- ▶ **Agregamos el codigo del saludo en el index**
- ▶ Arrancamos el servidor, junto con la aplicación
- ▶ Abrimos un navegador, con la dirección del servidor

```
<h1>Hola Blogolico, esto es Rails!</h1>
```



Primer aplicación

- ▶ Creamos el directorio para nuestras aplicaciones
- ▶ Creamos nuestra aplicacion rails
- ▶ Accedemos al directorio
- ▶ Creamos la base de datos
- ▶ Creamos el controlador para saludar
- ▶ Editamos la vista del controlador
- ▶ Agregamos el codigo del saludo en el index
- ▶ **Arrancamos el servidor, junto con la aplicación**
- ▶ Abrimos un navegador, con la dirección del servidor

```
ruby script/server
```



Primer aplicación

- ▶ Creamos el directorio para nuestras aplicaciones
- ▶ Creamos nuestra aplicacion rails
- ▶ Accedemos al directorio
- ▶ Creamos la base de datos
- ▶ Creamos el controlador para saludar
- ▶ Editamos la vista del controlador
- ▶ Agregamos el codigo del saludo en el index
- ▶ Arrancamos el servidor, junto con la aplicación
- ▶ **Abrimos un navegador, con la dirección del servidor**

```
firefox http://localhost:3000/saludo/index
```

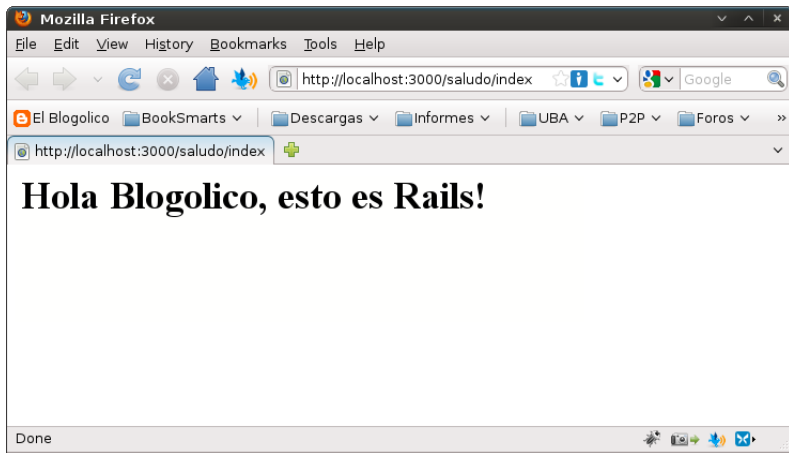


Primer aplicación

- ▶ Creamos el directorio para nuestras aplicaciones
- ▶ Creamos nuestra aplicacion rails
- ▶ Accedemos al directorio
- ▶ Creamos la base de datos
- ▶ Creamos el controlador para saludar
- ▶ Editamos la vista del controlador
- ▶ Agregamos el codigo del saludo en el index
- ▶ Arrancamos el servidor, junto con la aplicación
- ▶ Abrimos un navegador, con la dirección del servidor



Aplicación funcionando



Segunda aplicación

Partimos de nuestra reciente aplicación

- ▶ **Como segunda aplicación vamos a crear un Blog**
- ▶ Agregar, Editar, Mostrar y Borrar posts
- ▶ Scaffold

Segunda aplicación

Partimos de nuestra reciente aplicación

- ▶ Como segunda aplicación vamos a crear un Blog
- ▶ **Agregar, Editar, Mostrar y Borrar posts**
- ▶ Scaffold



Segunda aplicación

Partimos de nuestra reciente aplicación

- ▶ Como segunda aplicación vamos a crear un Blog
- ▶ Agregar, Editar, Mostrar y Borrar posts
- ▶ **Scaffold**

*Forma rápida de generar algunas de las principales piezas de una aplicación.
Modelos, vistas, controladores.*



Segunda aplicación

Partimos de nuestra reciente aplicación

- ▶ Como segunda aplicación vamos a crear un Blog
- ▶ Agregar, Editar, Mostrar y Borrar posts
- ▶ Scaffold



Segunda aplicación

- ▶ **Generamos la clase Post**
- ▶ Aplicamos los cambios en la base de datos
- ▶ Agregamos un link hacia el blog
- ▶ Entramos a nuestra aplicación
- ▶ Accedemos al Blog

```
ruby script/generate scaffold Post name:string title:string content:text
```



Segunda aplicación

- ▶ Generamos la clase Post
- ▶ **Aplicamos los cambios en la base de datos**
- ▶ Agregamos un link hacia el blog
- ▶ Entramos a nuestra aplicación
- ▶ Accedemos al Blog

```
rake db:migrate
```



Segunda aplicación

- ▶ Generamos la clase Post
- ▶ Aplicamos los cambios en la base de datos
- ▶ **Agregamos un link hacia el blog**
- ▶ Entramos a nuestra aplicación
- ▶ Accedemos al Blog

```
echo '<%= link_to "El Blog", posts_path %>' >> app/saludo/index.html.erb
```



Segunda aplicación

- ▶ Generamos la clase Post
- ▶ Aplicamos los cambios en la base de datos
- ▶ Agregamos un link hacia el blog
- ▶ **Entramos a nuestra aplicación**
- ▶ Accedemos al Blog

```
firefox http://localhost:3000/saludo/index
```



Segunda aplicación

- ▶ Generamos la clase Post
- ▶ Aplicamos los cambios en la base de datos
- ▶ Agregamos un link hacia el blog
- ▶ Entramos a nuestra aplicación
- ▶ **Accedemos al Blog**

*click en el link **El Blog***

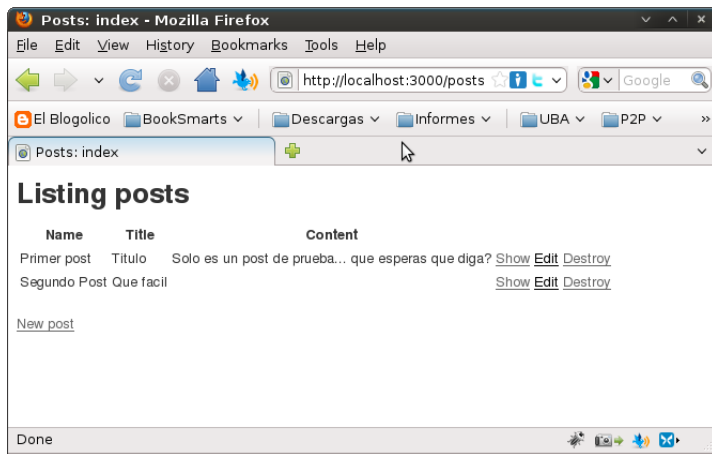


Segunda aplicación

- ▶ Generamos la clase Post
- ▶ Aplicamos los cambios en la base de datos
- ▶ Agregamos un link hacia el blog
- ▶ Entramos a nuestra aplicación
- ▶ Accedemos al Blog



Aplicación funcionando



Links Ruby

- ▶ <http://www.ruby-lang.org/es/>
- ▶ <http://www.ruby-doc.org/core/>
- ▶ <http://www.rubular.com/>
- ▶ <http://shoooes.net/>



Links Ruby

- ▶ <http://www.ruby-lang.org/es/>
- ▶ <http://www.ruby-doc.org/core/>
- ▶ <http://www.rubular.com/>
- ▶ <http://shoooes.net/>



Links Ruby

- ▶ <http://www.ruby-lang.org/es/>
- ▶ <http://www.ruby-doc.org/core/>
- ▶ <http://www.rubular.com/>
- ▶ <http://shoooes.net/>



Links Ruby

- ▶ <http://www.ruby-lang.org/es/>
- ▶ <http://www.ruby-doc.org/core/>
- ▶ <http://www.rubular.com/>
- ▶ <http://shoooes.net/>



Links Rails

- ▶ <http://rubyonrails.org/>
- ▶ <http://wiki.rubyonrails.org/>
- ▶ <http://railscasts.com/>
- ▶ <http://guides.rubyonrails.org/index.html>



Links Rails

- ▶ <http://rubyonrails.org/>
- ▶ <http://wiki.rubyonrails.org/>
- ▶ <http://railscasts.com/>
- ▶ <http://guides.rubyonrails.org/index.html>



Links Rails

- ▶ <http://rubyonrails.org/>
- ▶ <http://wiki.rubyonrails.org/>
- ▶ <http://railscasts.com/>
- ▶ <http://guides.rubyonrails.org/index.html>



Links Rails

- ▶ <http://rubyonrails.org/>
- ▶ <http://wiki.rubyonrails.org/>
- ▶ <http://railscasts.com/>
- ▶ <http://guides.rubyonrails.org/index.html>



Fin!

Fin!