

Programación con Ruby on Rails

Primeros Pasos

Delio Tolivia





Indice

Construir una nueva aplicación
Ejercicio

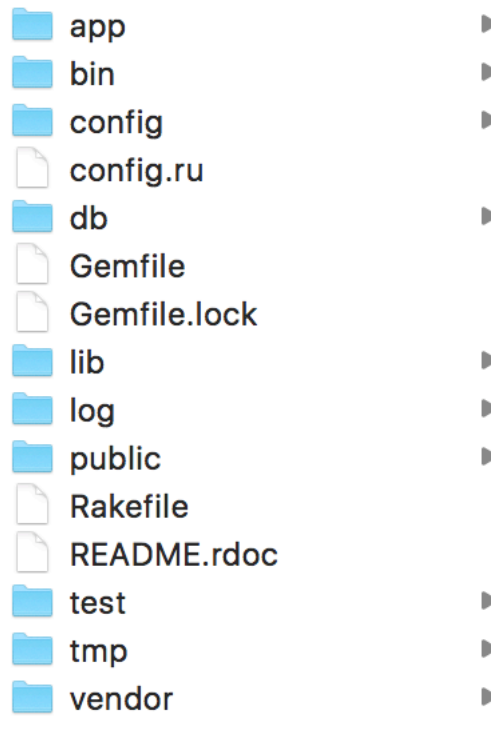
Creando una nueva aplicación (I)

- Al tener instalado Rails. Disponemos de una herramienta de comando: ***rails***
- Con este comando podremos crear nuestros nuevos proyectos de tal forma que se genera la estructura de directorios y ficheros necesarios.
- Para crear nuestra primera aplicación creamos una carpeta donde queramos ir colocando nuestros proyectos y desde la consola navegamos a ella. Creamos el proyecto demo con el comando:

rails new demo

Creando una nueva aplicación (II)

- Este comando crea un directorio llamado demo con varios subdirectorios y archivos en su interior



Creando una nueva aplicación (III)

- Podemos ver el estado de nuestra instalación con el comando

Rake about

- Con el podemos ver las versiones de las distintas herramientas y si hay algún problema para poder continuar
- Para arrancar el servidor de rails y ver nuestra aplicación usaremos el comando:

rails server

Creando una nueva aplicación (IV)

- El comando anterior creará un servidor web en el puerto 3000 en localhost. En esa ventana del terminal iremos viendo las acciones que realiza el servidor. Quedará arrancado y cuando queramos pararlo utilizaremos Ctrl-C.
- Por lo tanto desde un navegador ya podremos acceder mediante la url:

<http://localhost:3000>

Creando una nueva aplicación (V)

- Para nuestra primera aplicación no vamos a necesitar datos (Modelo). Por lo que empezaremos creando solamente un controlador. Para ello utilizaremos el comando:

rails generate controller Say hello goodbye

- Este comando nos creará la clase controlador SayController con los métodos ***hello*** y ***goodbye***. Su código lo podemos ver en:

apps/controllers/say_controller.rb

Creando una nueva aplicación (VI)

```
class SayController < ApplicationController  
  def hello  
  end  
  
  def goodbye  
  end  
end
```

- Vemos como nuestro controlador hereda de ***ApplicationController***.
- Para acceder a nuestros métodos desde el navegador utilizaremos las url:

http://localhost:3000/NOMBRE_CONTROLADOR/NOMBRE_METODO

<http://localhost:3000/say/hello>

Creando una nueva aplicación (VII)

- Tal y como vemos Rails ya ha enlazado la URL a la llamada a nuestros métodos del controlador y nos indica la vista relacionada para que podamos modificarla.

app/views/say/hello.html.erb

- Podemos modificar dicho fichero incluyendo html

<h1> Hello from Rails!</h1>

- Al volver a la URL podremos ver ya esta página que hemos realizado.

Creando una nueva aplicación (VIII)

- Como hemos visto nuestra vista se llama hello.html.erb en ella hemos escrito html pero también podemos incluir código en Ruby (por eso la extensión .erb).
- El código en estos ficheros se incluye entre las etiquetas `<%= %>` siendo interpretado y ejecutado antes de mostrar la página (si no es código que vaya a generar contenido en el html directamente puede ir en `<%.....%>`)
- Vamos a ampliar nuestra página y vamos a mostrar un contenido dinámico. Vamos a mostrar la hora actual en nuestra web. Para ello debemos modificar nuestro controlador para que calcule la hora y la vista para que la muestre.
- Para ello primero modificamos el método hello de nuestro controlador añadiendo la variable de instancia `@time` que almacene dicho dato:

```
def hello  
  @time = Time.now  
end
```

Creando una nueva aplicación (IX)

- Para mostrar este dato modificaremos nuestra vista hello.html.erb:

```
<h1> Hello from Rails!</h1>  
<p>  
  It is now <%= @time %>  
</p>
```

- Accediendo a la url de nuevo veremos la hora en nuestra página.

Creando una nueva aplicación (X)

- Tenemos otro método goodbye asociado a la vista goodbye.html.erb vamos a modificar su contenido

```
<h1>Goodbye!</h1>
```

```
<p>
```

```
  It was nice having you here
```

```
</p>
```

- Ya podemos ver nuestra nueva página accediendo a la url

<http://localhost:3000/say/goodbye>

Creando una nueva aplicación (XI)

- Ahora vamos a enlazar ambas ventanas. Para ello rails dispone de multiples “helper methods” que pueden usarse en las vistas. Vamos a utilizar el método ***link_to()*** que crea un hiperenlace. Modificamos nuestra vista `hello.html.erb`

```
<h1>Hello form Rails!</h1>
<p>
  It is now <%= @time =>
</p>
<p>
  Time to say
  <%= link_to “Goodbye”, say_goodbye_path %>!
</p>
```

- El primer parámetro es el texto que se mostrará en el enlace. El siguiente indica a rails que genere un enlace a la acción ***goodbye*** de del controlador ***say***

Creando una nueva aplicación (XII)

- Si vamos a nuestra url ya aparecerá el enlace a la página goodbye. Por lo tanto vamos a modificarla para colocar un enlace en ella a hello. De la misma manera incluiremos:

<p>

Say <%= link_to “Hello”, say_hello_path %> again.

</p>

- Con esto ya tenemos también el enlace a nuestra pagina hello desde goodbye

Ejercicio

- Intentar mostrar en nuestra página de hello la lista de ficheros
- Con el comando de Ruby ***Dir.glob***("*") nos devuelve un array con los ficheros del directorio actual.
- Usar un bucle ***for*** por ejemplo para recorrer el array en la vista.
- Usar elementos ****** para cada uno de los ficheros