

# Como empezar con Rails 5

En esta guía te explicamos los pasos necesarios para empezar a crear una aplicación Rails 5 y alojarla en uno de nuestros planes de hosting.

Ten en cuenta que debes [acceder mediante SSH a tu cuenta de hosting](#) y ejecutar algunos comandos.

## 1.- Crea aplicación Ruby desde el panel de hosting

Accede al panel de Hosting y luego pincha sobre el icono **Aplicaciones Ruby**. Una vez allí rellena y envía el formulario, tal y como se ve en la siguiente imagen:



The screenshot shows a web interface for creating a Ruby application. At the top, there is a breadcrumb trail: Home > Hosting > Plan Developer midominio123.com > Panel de Hosting > Aplicaciones Ruby. The main heading is 'Aplicaciones Ruby' with a green 'MOSTRAR AYUDA' button. Below this is a section titled 'Crear nueva aplicación Ruby'. The form contains three input fields: 'Nombre de la aplicación:' with the value 'miapp' and a note 'Se usará como nombre de directorio de la aplicación, nada de caracteres raros.'; 'Dominio:' with a dropdown menu showing 'midominio123.com / f1a38f17.gclientes.com'; and 'Stack:' with a dropdown menu showing 'Ruby Kon (Ruby 2.2 + Rubygems 2.4.x)' and a note 'Se permite un único stack por cuenta de hosting. Una vez creado una app, el resto usarán el mismo stack'. At the bottom, there are two buttons: 'Cancelar' (grey) and 'Enviar' (purple).

En este caso, poniendo **miapp** como nombre de la aplicación, la ruta completa de la aplicación será **/home/usuario/ruby/miapp**, donde **usuario** es el nombre de usuario de tu hosting.

## 2.- Instala Rails y Bundler

Instalamos la versión 5.0 de Rails:

```
<br />  
gem install rails -v5.0<br />
```

En principio, al instalar Rails 5.0 también se habrá instalado la última versión compatible de Bundler. Si por algún motivo no se ha instalado, lo puedes instalar ejecutando:

Y luego instalamos Bundler:

```
<br />  
gem install bundler<br />
```

### 3.- Crea la aplicación Rails en tu hosting

Al configurar la nueva aplicación desde el panel de hosting, te habremos indicado cual es el directorio donde debes ubicar los archivos de la misma. En este caso, el directorio de la aplicación se debe llamar **miapp**.

Accede vía SSH y ejecuta:

```
<br />  
cd ruby<br />  
rm -fr miapp<br />  
rails new miapp<br />
```

Y seguido entramos al directorio de nuestra aplicación:

```
<br />  
cd miapp<br />
```

### 4.- Edita el archivo Gemfile

Ahora procederemos a realizar las modificaciones en el fichero Gemfile según las necesidades de nuestro proyecto.

Para hacer esto puedes usar el editor *nano*:

```
<br />  
nano Gemfile<br />
```

Para **grabar** el archivo **Gemfile** presionas **CTRL O**, y este se grabara, luego para **salir** del editor presionas **CTRL X**

Ahora puedes elegir entre hacer uso de la gema The Ruby Racer o bien execjs. Recomendamos la segunda de las opciones.

## **The Ruby Racer (Opcional)**

En este archivo descomentamos la linea que dice:

```
<br />  
#gem 'therubyracer', platforms: :ruby<br />
```

Donde quitaremos el simbolo #, dejandolo así:

```
<br />  
gem 'therubyracer', platforms: :ruby<br />
```

## **NodeJs - execjs (Opcional)**

[Como usar nodejs en tu aplicación Ruby a través de la gema execjs](#)

## **5.- Instala dependencias con Bundler**

Ahora tendrás que instalar las gemas necesarias para tu aplicación, para ello ejecuta:

```
<br />  
bundle install<br />
```

## **6.- Define el secret\_key\_base**

Lo primero es obtener una cadena suficientemente larga y aleatoria, para ello ejecutamos:

```
<br />  
rake secret<br />
```

Se nos mostrará la cadena que usaremos como **secret\_key\_base** de la aplicación. Ahora abrimos el archivo **config/secrets.yml**:

```
<br />  
nano config/secrets.yml<br />
```

Y pegamos nuestra cadena en la sección **production** del archivo.

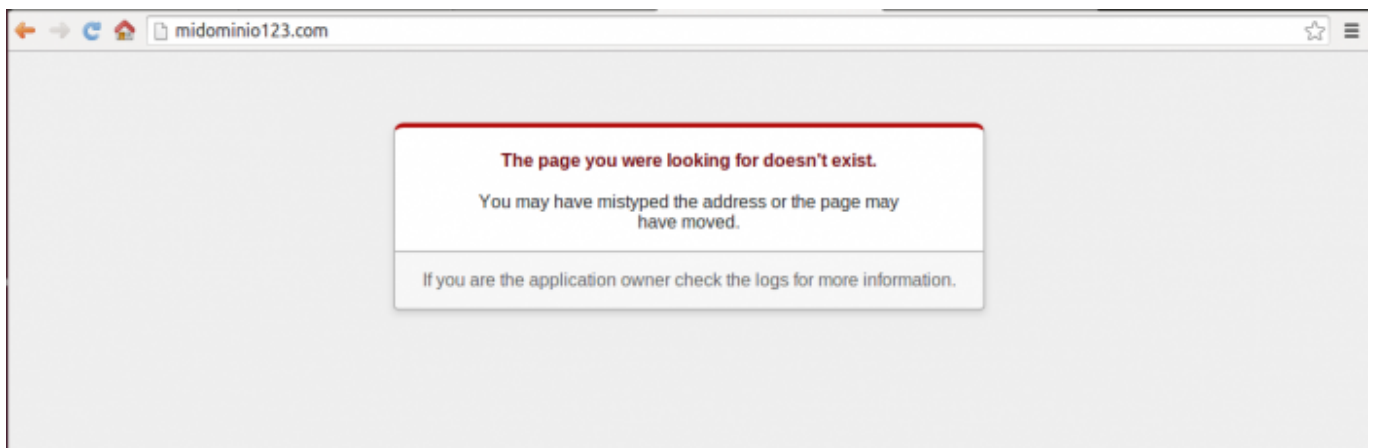
## 7.- Reinicia la aplicación

Simplemente ejecuta:

```
<br />  
touch tmp/restart.txt<br />
```

## 8.- Verifica el estado de la instalación

Para verificar el estado de la instalación nos vamos a nuestra pagina web, donde deberá mostrarnos una pantalla como la que se muestra a continuación:



## 9.- Genera el controlador de Bienvenida

Simplemente ejecuta:

```
<br />
```

```
rails generate controller welcome index<br />
```

## 10.- Personalizar el mensaje de bienvenida

Abrimos el archivo de vista **app/views/welcome/index.html.erb** generado en el paso anterior:

```
<br />  
nano app/views/welcome/index.html.erb<br />
```

Borramos todo el contenido del archivo y colocamos solamente:

```
</p>  
<h1>Hola, Rails!</h1>  
<p>
```

En el editor *nano*, para **grabar** el archivo Gemfile presionas **CTRL O**, y este se grabara, luego para **salir** del editor presionas **CTRL X**

## 11.- Configura ruta para bienvenida

Vamos a hacer que Rails muestre en la raíz de la web, es decir, en «/», el controlador y vista generados en los pasos previos.

Para ello abrimos el archivo config/routes.rb :

```
<br />  
nano config/routes.rb<br />
```

Y añadimos la siguiente línea antes del operador **end**

```
<br />  
root to: &#8216;welcome#index&#8217;<br />
```

El archivo podría quedar así:

```
<br />  
Rails.application.routes.draw do<br />  
  root to: &#8216;welcome#index&#8217;</p>
```

```
<p> # For details on the DSL available within this file, see  
http://guides.rubyonrails.org/routing.html<br />  
end<br />
```

En el editor Nano, para **grabar** el archivo Gemfile presionas **CTRL O**, y este se grabara, luego para **salir** del editor presionas **CTRL X**

## 12.- Reinicia la aplicación

Simplemente ejecuta:

```
<br />  
touch tmp/restart.txt<br />
```

## 13.- Accede a la web

Finalmente, después de haber modificado el contenido del controlador de Bienvenida, verificamos en nuestra pagina Web

