

Como empezar con Rails 5.2 y ruby 2.6

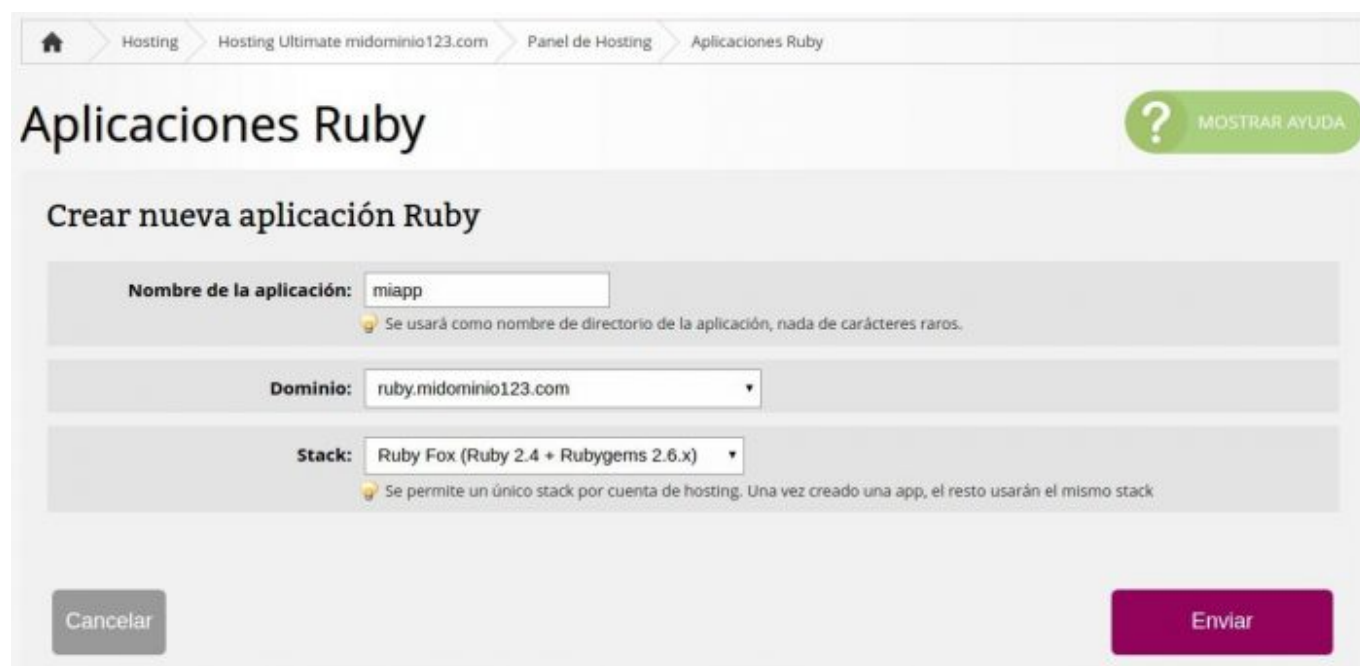
En esta guía te explicamos los pasos necesarios para empezar a crear una aplicación Rails 5.2 y alojarla en uno de nuestros planes de hosting con ruby 2.6, disponible desde enero de 2019.

Para hacer uso de este grupo de software, tienes que tener en cuenta que si tu plan fue creado en guebs.com hace más de 1 año es conveniente que te pongas en contacto con nosotros. Usar ruby 2.6 y rails 5.2 requiere de las capacidades de nuevo software. Proporcionaremos a tu cuenta estas características para que pueda usarlas.

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:

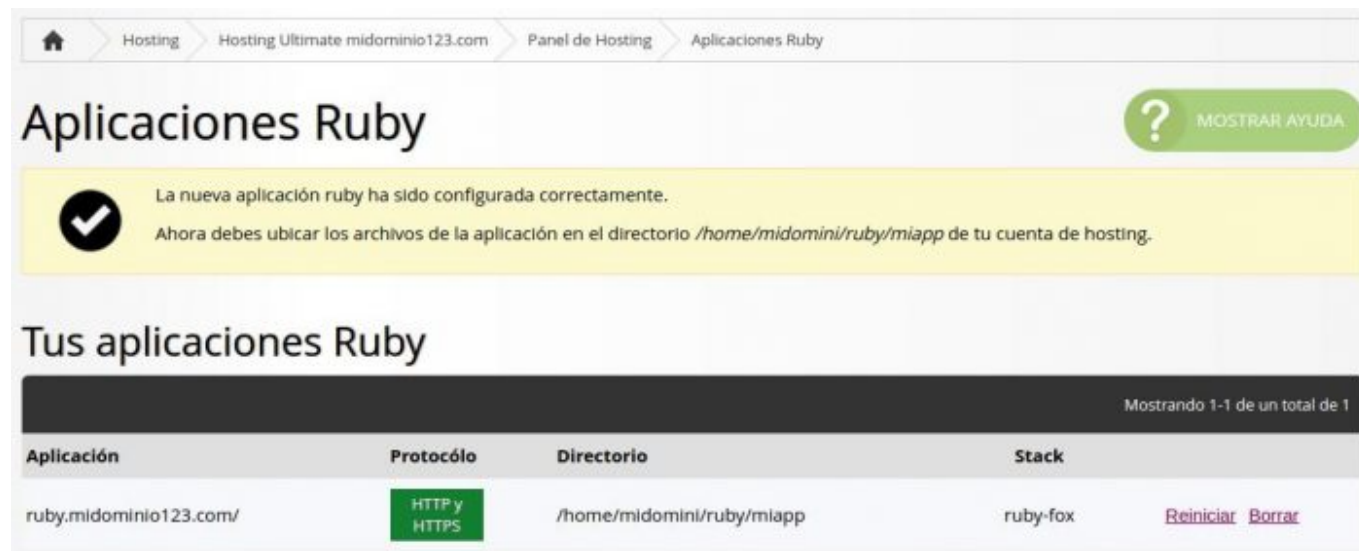


The screenshot shows a web interface for creating a Ruby application. At the top, there is a breadcrumb trail: Home > Hosting > Hosting Ultimate midominio123.com > Panel de Hosting > Aplicaciones Ruby. The main heading is 'Aplicaciones Ruby' with a green 'MOSTRAR AYUDA' button. Below this is the section 'Crear nueva aplicación Ruby'. The form has three rows:

- Nombre de la aplicación:** A text input field containing 'miapp'. Below it is a note: 'Se usará como nombre de directorio de la aplicación, nada de caracteres raros.'
- Dominio:** A dropdown menu showing 'ruby.midominio123.com'.
- Stack:** A dropdown menu showing 'Ruby Fox (Ruby 2.4 + Rubygems 2.6.x)'. Below it is 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 left is a 'Cancelar' button and at the bottom right is a purple 'Enviar' button.

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.



The screenshot shows a web interface for managing Ruby applications on a hosting service. At the top, there is a breadcrumb trail: Home > Hosting > Hosting Ultimate midominio123.com > Panel de Hosting > Aplicaciones Ruby. The main heading is "Aplicaciones Ruby" with a green "MOSTRAR AYUDA" button. A yellow notification banner contains a checkmark icon and the text: "La nueva aplicación ruby ha sido configurada correctamente. Ahora debes ubicar los archivos de la aplicación en el directorio /home/midomini/ruby/miapp de tu cuenta de hosting." Below this is a section titled "Tus aplicaciones Ruby" with a sub-header "Mostrando 1-1 de un total de 1". A table lists the application details:

Aplicación	Protocolo	Directorio	Stack	
ruby.midominio123.com/	HTTP y HTTPS	/home/midomini/ruby/miapp	ruby-fox	Reiniciar Borrar

2.- Instala Rails y Bundler

Ahora necesitamos acceder via SSH para empezar a instalar nuestra aplicación Rails 5.2.x

Para instalar la versión 5.2.x de Rails ejecutamos:

```
<br />  
gem install rails -v5.2<br />
```

En principio, al instalar Rails 5.2 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**.

vamos a acceder al directorio «*ruby*», borrar el directorio «*miapp*» y crear una nueva instalación de rails con el nombre «*miapp*» y así volver a crear el directorio.

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

El comando anterior crea la aplicación con mysql como base de datos. Si quieres usar sqlite, puedes eliminar la opción «-d mysql» y la aplicación se generará con sqlite automáticamente.

Si quieres que tu aplicación use postgres, entonces tienes que añadir «-d postgresql»

```
<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 debemos configurar la gema ExecJs y editar el primer external Runtime

```
</p>  
<p>nano  
~/ruby/gems/gems/execjs-2.7.0/lib/execjs/runtimes.rb</p>  
<p>  
  
<br />  
Node = ExternalRuntime.new(<br />  
name: Node.js (V8),<br />  
# command: [nodejs, node],<br />  
command: /opt/nodejs-10/bin/node,<br />  
runner_path: ExecJS.root + /support/node_runner.js,<br />  
encoding: &#8216;UTF-8&#8217;<br />  
)<br />
```

Como usar nodejs en tu aplicación Ruby a través de la gema execjs

7.- Reinicia la aplicación

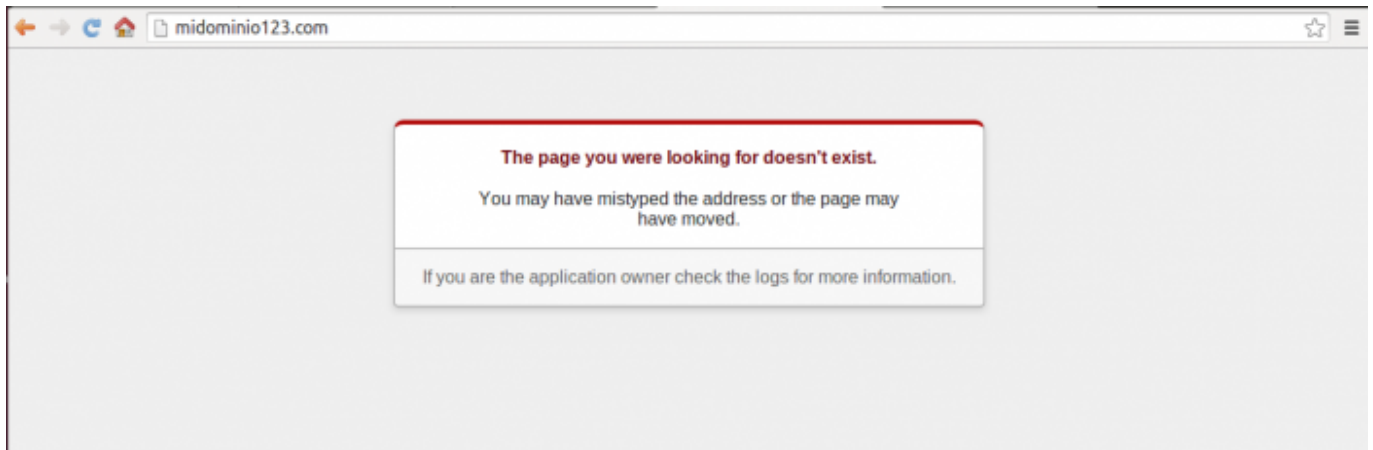
Simplemente ejecuta:

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

Opcionalmente, puedes reiniciar la aplicación desde el panel de hosting de guebs.

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

El archivo podría quedar así:

```
<br />  
Rails.application.routes.draw do<br />  
  get &#8216;welcome/index&#8217;</p>  
<p>root &#8216;welcome#index&#8217;<br />  
</p>  
<p># For details on the DSL available within this file, see  
<a href="http://guides.rubyonrails.org/routing.html">http://guides.rubyonrails.org/routing.html</a><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.- Precompilar assets

Ejecutamos la precompilacion de los recursos como CSS:

```
<br />  
RAILS_ENV=production bundle exec rake assets:precompile<br />
```

13.- Reinicia la aplicación

Simplemente ejecuta:

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

Opcionalmente, puedes reiniciar la aplicación desde el panel de hosting de guebs.

14.- Accede a la web

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



15.- Configura una base de datos mysql en tu aplicación

Primero debemos crear la base de datos y el usuario. Podemos hacerlo desde el panel de hosting.

Como crear una base de datos MySQL

Debemos configurar el archivo **config/database.yml** de la aplicación entonces . Al final encontraremos la zona de «production:» . Podemos optar por escribir el password en el archivo, o bien por escribir el password en una variable que será leída por la aplicación. Una «env»

Si elegimos esta segunda opción (donde pone 'MIAPP' hay que cambiarlo por el nombre de tu aplicación) :

```
</p>
<p>echo                                &#8216;export
MIAPP_DATABASE_PASSWORD=password_de_la_base_de_datos&#8217;
&gt;&gt; ~/.bashrc</p>
<p>
```

La zona indicada debe quedar de este modo:

```
production:
<<: *default
```

```
database: nombre_base_de_datos  
username: usuario_base_datos  
password: <%= ENV['MIAPP_DATABASE_PASSWORD'] %>
```

Con esto la base de datos ya está configurada.