**Administración de usuarios en ruby**

1. [Path: Setting Things Up](http://www.peoplecancode.com/en/paths/1)

Device es una gema extraordinaria que nos ahorrará mucho trabajo. La utilizaremos para permitir que los usuarios creen cuentas en nuestra página Web, modifiquen sus perfiles, inicien o cierren su sesión, o también que les enviemos recordatorios de sus contraseñas y más. Todo esto es seguro y fácil de configurar. Podemos poner a mano todas las funcionalidades que Device proporciona, pero no lo haremos por dos razones: Primero, toma mucho tiempo y en segundo, el nivel de arreglo que necesitamos para nuestro proyecto no es tan complicado. Por lo tanto, una disposición estándar Device es más que suficiente.

Paso 1: Lee Devise's Github y agrega Gem Devise

Vayamos a la página de [Devise's github page](https://github.com/plataformatec/devise).Te recomiendo que leas la documentación. Todo lo que necesitas saber está allí. En los próximos tutoriales, modificaremos Device así que necesitaremos regresar a esta página. Por ahora, toma tu tiempo y chécalo.

Primero, necesitamos agregar Device a nuestro Gemfile. El Gemfile es la lista de todos los “plugins” o como los llamamos en la comunidad de Ruby “gems.” Gems son los mini-programas autónomos que se construyen en una manera que permita que las combinemos fácilmente con nuestra aplicación.

## Paso 2: Agrupa las Gemas

Ahora necesitamos agrupar Device con nuestra aplicación. Eso significa que el encargado de la gema - llamado bundler - descargará todas las gemas necesarias del Internet. Este proceso puede tomar un rato dependiendo del tamaño de la gema o si es la primera vez que lo haces.

## Paso 3: Genera el archivo de Configuration

En el siguiente paso necesitamos generar un archivo de configuración para Devise.

## Paso 4.1: Configura los archivos de Environments

Veamos paso por paso las instrucciones del mensaje:

Dentro del archivo de config/environments hay otros tres archivos. Uno por cada entorno. Cada uno se puede establecer con diferentes configuraciones.

En la línea de Devise nos está proponiendo que debemos ir dentro de  development.rb y test.rb

config/environments/development.rb config/environments/test.rb

config.action\_mailer.default\_url\_options = { host: 'localhost:3000' }

Dentro de nuestro archivo de production.rb necesitamos poner un verdadero URL que empareje nuestro dominio. Si estás utilizando Heroku, éste sería el nombre de dominio de Heroku, por ejemplo: http://your-demo-application.herokuapp.com. Si ya has conectado tu heroku con un dominio de encargo, este sería http://www.custom-domain.com.

Esto es importante porque todos los emails que Devise le esté enviando a los usuarios - reajustes de contraseña, confirmaciones de cuenta, etc. - señalarán a esta dirección.

config/environments/development.rb config/environments/production.rb

## Paso 4.3: Crea "Static Pages"

1. Si sigues el tutorial completo de  ["creating static pages tutorial"](http://www.peoplecancode.com/tutorials/how-to-create-static-pages-in-ruby-on-rails-application) ya estás listo. Sino, regresa y has ese [tutorial](http://www.peoplecancode.com/tutorials/how-to-create-static-pages-in-ruby-on-rails-application).

## Paso 4.4: Agrega los Mensjes Flash

3. Las aplicaciones de Rails usan mensajes de flash para notificarnos sobre acontecimientos. Nosotros usaremos muchos mensajes de flash en el futuro, así que queremos que tengan buen aspecto. El ejemplo propuesto por Devise funcionara, pero pronto llegará a ser insuficiente para nuestra aplicación. Utilizaremos un código diferente.

## Paso 5: Configura los correos electrónicos

Ya casi terminamos con la configuración. La última cosa que tenemos que establecer es el sistema de correo electrónico para que nuestra aplicación pueda enviar correos electrónicos, como para los de restablecer contraseñas.

## Paso 6: Modifica la configuración de producción y desarrollo

A continuación, agregaremos una configuración para enviar mensajes de correo electrónico reales. En la carpeta config/environments modificaremos ambas configuraciones development y production:

/config/environments/development.rb

## Paso 7: Ejecuta Rails Server

En este punto, es una buena idea reiniciar la aplicación. El generador que acabamos de correr, ha creado un nuevo archivo inicializador. Todos los inicializadores se cargan durante el inicio de la aplicación, así que si estás ejecutando tu servidor en este momento hazlo comenzar de nuevo. En la terminal presiona en el teclado "control + C" y escribe:

$ rails server

## Paso 8: Checa a la base de datos y ejecuta "Rake Routes"

El primer archivo es un archivo de migración: 20140506090012\_devise\_create\_users.rb piensa en el como una instrucción para hacer cambios en la base de datos. Utilizaremos nuestra base de datos por primera vez. Abre el archivo checa lo que dice:

db/migrate/20140506090012\_devise\_create\_users.rb

## Paso 9: Añade el enlace al Master Layout

Ahora, vamos a utilizar las rutas y ponerles un enlace para iniciar la sesión en nuestro diseño maestro:

/app/views/layouts/application.html.erb