

SISTEMA USUARIOS.

Generaremos una aplicación para ejemplificar la implementación de formularios en una aplicación.

Tenemos un sistema de registro de usuarios administrado por una entidad “administrador”.

Por ahora sólo haremos la creación de los modelos asociados a la entidad “usuarios” y la entidad “administrador”.

GENERAR EL PROYECTO.

Ya teniendo instalado ruby on rails generamos la estructura de nuestra aplicación:

```
$ rails new SistemaUsuarios
```

Esta instrucción nos genera la carpeta *SistemaUsuarios* en ella encontraremos una estructura de directorios que contendrán la construcción de nuestro sistema conforme al patrón de diseño MVC.

Tendremos dos entidades:

- * “Administrador” : Un administrador tendrá un Nombre, un correo electrónico y una contraseña.

- * “Usuario”: Un usuario tendrá un nombre, correo electrónico, dirección, teléfono y edad.

Generaremos los modelos asociados a cada una de las entidades ejecutando estas instrucciones desde la consola posicionados en la carpeta del proyecto:

```
rails generate model User name:string email:string address:string phone:string age:integer
```

```
rails generate model Admin name:string email:string password:string
```

En estas instrucciones especificamos los atributos de estas entidades y el tipo de dato que recibirán . Nos genera además una serie de archivos en nuestro proyecto, en especial un archivo que se guarda en la carpeta *db/migrate* que contiene la estructura de nuestro modelo o “tabla”.

Al ejecutar la instrucción `rake db:migrate` modificamos el archivo `schema.rb` que contiene la estructura de los modelos, es decir, de la base de datos de nuestro proyecto conforme a las migraciones que generamos con la instrucción `rails generate model`:

```
clau@claus:~/Escritorio/Ayudantias/IS2/RAILS/SistemaUsuarios$ rails generate model User name:string email:string address:string phone:string age:integer
  invoke  active_record
  create  db/migrate/20150927015100_create_users.rb
  create  app/models/user.rb
  invoke  test_unit
  create  test/models/user_test.rb
  create  test/fixtures/users.yml
clau@claus:~/Escritorio/Ayudantias/IS2/RAILS/SistemaUsuarios$ rails generate model Admin name:string email:string password:string
  invoke  active_record
  create  db/migrate/20150927015107_create_admins.rb
  create  app/models/admin.rb
  invoke  test_unit
  create  test/models/admin_test.rb
  create  test/fixtures/admins.yml
clau@claus:~/Escritorio/Ayudantias/IS2/RAILS/SistemaUsuarios$
clau@claus:~/Escritorio/Ayudantias/IS2/RAILS/SistemaUsuarios$ rake db:migrate
== 20150927015100 CreateUsers: migrating =====
-- create_table(:users)
   => 0.0023s
== 20150927015100 CreateUsers: migrated (0.0025s) =====
== 20150927015107 CreateAdmins: migrating =====
-- create_table(:admins)
   => 0.0011s
== 20150927015107 CreateAdmins: migrated (0.0012s) =====
clau@claus:~/Escritorio/Ayudantias/IS2/RAILS/SistemaUsuarios$
```

Si las migraciones son correctas vemos en la consola la creación de las tablas.

Ya tenemos una pequeña base de datos donde podemos operar.

Como ya hemos visto en clase, un controlador es el conector entre la vista que es presentada al usuario y los modelos que representan las tablas en la base de datos.

Respetando el patrón de diseño, necesitamos especificar los requerimientos de cada una de las entidades para que sea válido y poder guardarlo en la base de datos.

GENERAR CONTROLADOR

Generamos un nuevo controlador con las acciones básicas de un CRUD: crear, ver, actualizar y borrar.

Esto lo obtenemos con la instrucción *rails generate controller* de la siguiente forma:

rails generate controller CRUD index create read update delete

```
clau@claus:~/Escritorio/Ayudantias/IS2/RAILS/SistemaUsuarios$ rails generate controller CRUD index create read update delete
   create  app/controllers/crud_controller.rb
   route   get 'crud/delete'
   route   get 'crud/update'
   route   get 'crud/read'
   route   get 'crud/create'
   route   get 'crud/index'
   invoke  erb
   create  app/views/crud
   create  app/views/crud/index.html.erb
   create  app/views/crud/create.html.erb
   create  app/views/crud/read.html.erb
   create  app/views/crud/update.html.erb
   create  app/views/crud/delete.html.erb
   invoke  test_unit
   create  test/controllers/crud_controller_test.rb
   invoke  helper
   create  app/helpers/crud_helper.rb
   invoke  test_unit
   invoke  assets
   invoke  coffee
   create  app/assets/javascripts/crud.coffee
   invoke  scss
   create  app/assets/stylesheets/crud.scss
clau@claus:~/Escritorio/Ayudantias/IS2/RAILS/SistemaUsuarios$
```

Nos genera una serie de modificaciones en el proyecto.

El archivo de nuestro controlador se genera en la carpeta *app/controller* con el nombre *crud_controller.rb*

route get 'crud/delete', route get 'crud/update', route get 'crud/read', route get 'crud/create' y route get 'crud/index' son modificaciones que se realizan en el archivo *config/routes.rb*, en el cual se especifica las rutas que vamos a poder tener acceso desde el navegador y con cada ruta podemos visualizar la vista asociada a cada acción del controlador, en este caso las vistas se generaron en la carpeta *app/views/crud*, el nombre de cada vista hace referencia a cada acción del controlador, el cual luce de la siguiente forma.

Entonces vamos a definir distintas cosas en cada una de las acciones:

- ❖ **index.** Presentar todos los registros de usuarios que existen en la base de datos.
- ❖ **create.** Generar un nuevo registro en la base de datos.
- ❖ **read.** Ver la información del registro desplegada.
- ❖ **delete.** Eliminar un usuario a partir de la selección de usuarios en el index.

Todo esto lo vamos a generar Active Record y su ORM que está definido en rails, el cual facilita la creación y uso de los objetos mediante un sistema de mapeo de objetos de forma relacional.

Consulta su uso en:

http://guides.rubyonrails.org/active_record_basics.html#active-record-as-an-orm-framework

Definimos la implementación de create index, update y delete en el repositorio:
git clone <https://github.com/Claudiarg/IS2.git>