

## Demo de Ruby on Rails

### 1. Requisitos

- 0. Tener Ruby + Rails instalado
- 1. Levantar instancia del mySQL.
- 2. Crear un usuario **ror**, con clave **ror**
- 3. Crear una base de datos 'eci\_development' y darle permisos sobre la misma al usuario **ror**

### 2. Demo

#### **CREAR APLICACION**

---

- 1. En el directorio seleccionado, crear estructura de directorios para la aplicación

```
# rails ECI
# cd ECI
```

- 2. Mostrar la estructura de directorios.
- 3. Abrir otra ventana de terminal e iniciar server:  

```
# ruby script/server
```
- 3. Probarlo en el browser  

```
=> http://0.0.0.0:3000/
```

#### **CONTROLLER**

---

- 1. Generar el controller para la inscripción a la ECI  

```
# ruby script/generate controller inscripcion
```
- 2. Mostrar los archivos generados

3. Probarlo en el browser y conseguir un error  
=> <http://0.0.0.0:3000/inscripcion>

4. Editar '**app/controllers/inscripcion\_controller.rb**' y agregar:

```
def index
  render :text => "hola ECI!"
end
```

5. Probarlo en el browser exitosamente  
=> <http://0.0.0.0:3000/inscripcion>

6. Borrar el método recién agregado.

## ***MODELO: INSCRIPTOS***

---

1. Crear el modelo para los inscriptos al curso  
# ruby script/generate/model inscripto

2. Mostrar los archivos generados

3. Editar el archivo **db/migrate/001\_create\_inscriptos.rb** y agregar las columnas:

```
create_table :inscriptos do |t|
  t.column "nombre", :string
  t.column "apellido", :string
  t.column "documento", :string
end
```

-- *Vamos a usar una base de datos MySql que ya tenemos creada.*

4. Abrir el navegador de MySql y mostrar la base vacía

5. Editar el entorno de la base **config/database.yml**

```
development:
  adapter: mysql
  database: eci_development
  username: ror
  password: ror
  socket: /tmp/mysql.sock
```

6. Correr la migración para crear la tabla “inscriptos”:  
# rake db:migrate

7. Mostrar que creó la tabla “inscriptos”. Mostrar los campos y destacar el campo “id”.

8. Mostrar que creó la tabla “schema\_info” y ver el número de versión



-- Siempre que modifiquemos la configuración de la base de datos debemos recordar reiniciar el servidor

9. Reiniciar el servidor.

## SCAFFOLDING IMPLÍCITO

---

-- Vamos a dejar que Rails nos provea una primera implementación de la administración de inscriptos.

1. Agregar en 'app/controllers/inscription\_controller.rb' :

```
scaffold :inscripto
```

2. Probar en el browser

=> <http://0.0.0.0:3000/inscripcion>

3. Agregar y editar un nuevo inscripto.

4. Resaltar el hecho de que el controller no tiene métodos explícitos y en la estructura de directorios no hay páginas HTML y sin embargo la aplicación funciona

## VALIDACIONES

---

1. Agregar en 'app/models/inscripto.rb':

```
validates_presence_of :documento
```

2. Probar de agregar un inscripto sin documento

## MODELO: CURSOS

---

-- Tenemos inscriptos pero no tenemos cursos!

1. Generar el modelo para los cursos:

```
# ruby script/generate model curso
```

2. Agregar en 'db/migrate/002\_create\_cursos.rb' la tabla "cursos" con los campos:

```
create_table :cursos do |t|
  t.column "titulo", :string
  t.column "orador", :string
```



```
t.column "inicio", :date
t.column "fin", :date
t.column "turno", :string
end
```

3. Agregar la referencia al curso correspondiente en la tabla de inscriptos:

```
add_column "inscriptos", "curso_id", :integer
```

4. Correr las migraciones para actualizar la base de datos:

```
# rake db:migrate
```

5. Mostrar las tablas actualizadas en la base de datos. Mostrar el número de versión del esquema.

6. Mostrar el archivo '**db/schema.db**' que mantiene la última versión del esquema.

## ASOCIACIONES

---

1. Ir al model de inscripto y agregar:

```
belongs_to :curso
```

2. Ir al model de curso y agregar:

```
has_many :inscriptos
```

3. Probar en el browser para verificar que todo sigue andando...

## SCAFFOLDING EXPLÍCITO

---

*-- Tenemos los inscriptos, tenemos los cursos y tenemos la asociación entre ambos. Aún así, la aplicación así como está no nos permite inscribir a alguien en un curso... Para poder llevar a cabo esta tarea que es un poco más compleja, vamos a cambiar de estrategia. En lugar de contar con el scaffolding automático de Rails, vamos a pedirle que nos genere todos los archivos en forma explícita para que podamos adaptarlos a nuestras necesidades.*

1. Generar el scaffold para los inscriptos:

```
# ruby script/generate scaffold inscripto inscripcion
```

2. Mostrar como cambio el controller y los templates que se generaron en **"app/views/inscripcion"**

3. Probar en el browser para verificar que sigue andando igual que antes

## VISTA

---

-- Vamos a modificar el formulario de alta de inscripción para que se pueda seleccionar el curso

1. Abrir **app/views/inscripcion/\_form.rhtml** y agregar info de curso al final:

```
<p><label for="inscripto_curso">Curso</label><br/>
<%= select 'inscripto', 'curso_id',
          Curso.find(:all).collect{|c|
[c.titulo,c.id]}%></p>
```

2. Probar en el browser

-- Pero no hay cursos en el combo!

## CONSOLA

---

1. En la línea de comando, iniciar una consola Rails:

```
# ruby script/console
```

-- En esta consola tenemos acceso a todas las clases de nuestra aplicación

2. Crear en forma interactiva un nuevo curso:

```
# >> c = Curso.create
# >> c.titulo = 'Ruby, una joya!'
# >> c.orador = 'ale y diego'
# >> c.save
```

3. Volver a probar en el browser

## DETALLE DE INSCRIPCIÓN

---

-- Finalmente, modifiquemos la vista del detalle de una inscripción para que muestre a curso corresponde

1. Abrir **'app/views/inscripcion/show.rhtml'** y agregar al final:

```
<p>
  <b>Curso:</b> <%=h @inscripto.curso.titulo %>
</p>
```

2. Probarlo en el browser  
OJO QUE LOS QUE NO TENGAN CURSO ASOCIADO VAN A DAR ERROR!

FIN.