

# PROYECTO FINAL DE DESARROLLO DE APLICACIONES WEB

---

## Introducción:

uDraw es una aplicación web que permite a los usuarios poder elaborar Modelos Entidad Relación usados en Bases de Datos, con la finalidad de que a través de esta herramienta, sus diseños sean más rápidos y eficientes.

## Objetivos del Proyecto

- Poner en práctica los conocimientos adquiridos en clase en el uso de las diferentes tecnologías web como son Ajax, HTML5, jquery.
- Poder integrar nuevos conocimientos aprendidos fuera del aula de clase sobre el uso de otra tecnología web lo es: Ruby on Rails, mysql, soap.

## Estructura del Proyecto

- El proyecto consta de la opción de registrar para los usuarios que formen parte de la comunidad politécnica con autenticación de Espol u otras cuentas de redes sociales.

The screenshot displays a web interface for user authentication, divided into two main sections: 'Registrarse' (Register) and 'Iniciar sesión' (Login). The 'Registrarse' section on the left includes input fields for 'Nombre:' (Name), 'Correo Electrónico:' (Email Address) with an example 'Ex: foo@gmail.com', and 'Contraseña:' (Password). Below these fields is a checkbox labeled 'Acepto los Términos de uso' (I accept the Terms of Use) and a blue 'Registrarse' button. The 'Iniciar sesión' section on the right features four social login buttons: 'Iniciar sesión con Facebook' (blue), 'Iniciar sesión con Twitter' (light blue), 'Iniciar sesión con Google' (red), and 'Iniciar sesión con su cuenta Espol' (green). The background of the interface is a blurred image of a person's face.

# Cuenta Espol



**Correo Electrónico:**


@espol.edu.ec

**Contraseña:**

[Ingresar](#)

- Una vez hecho el registro correspondiente podemos proceder a entrar a la aplicación. Si es un usuario nuevo entonces la vista de archivos estará vacía dado que no hay ningún archivo, de lo contrario estarán los archivos recientes y compartidos en la vista.

Archivo -
Crear nuevo modelo
Back


apinargo  
400 MB

Mis Archivos Recientes
Archivos Compartidos Conmigo

Diagram Fer

|          |
|----------|
| Entidad  |
| Atributo |

Diagrama Adriano

|          |
|----------|
| Entidad  |
| Atributo |

Diagrama Leonardo

Diagram Fer

|          |
|----------|
| Entidad  |
| Atributo |

Diagrama Adriano

|          |
|----------|
| Entidad  |
| Atributo |

Diagrama Leonardo

- Podemos acceder a nuestro perfil y modificar las cuentas usadas.

## Administrar Cuenta

**Name**

**Twitter**

**Facebook**

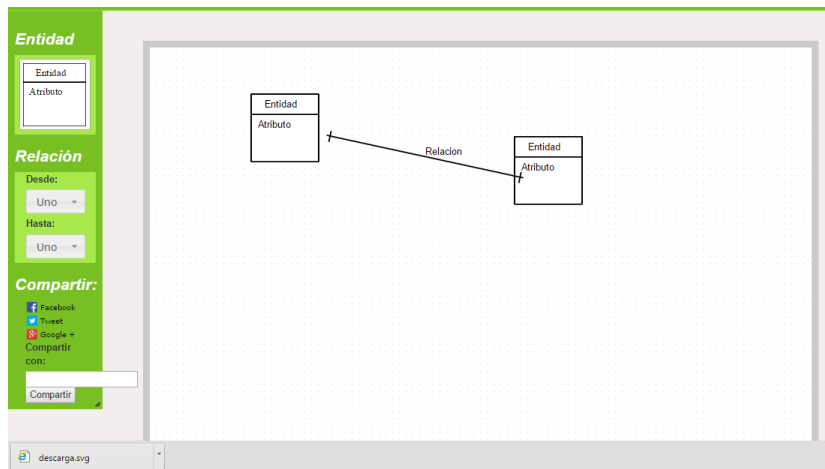
**Google**

**Espol**

[Update User](#)

[Back](#)

- Podemos crear modelos a nuestro gusto, compartirlos con otros usuarios, eliminarlos, exportarlos a nuestro computador, o simplemente almacenarlos en la base de datos.



CREAMOS UN  
DIAGRAMA QUE YA  
PUEDE SER  
EXPORTADO

**Guardar**

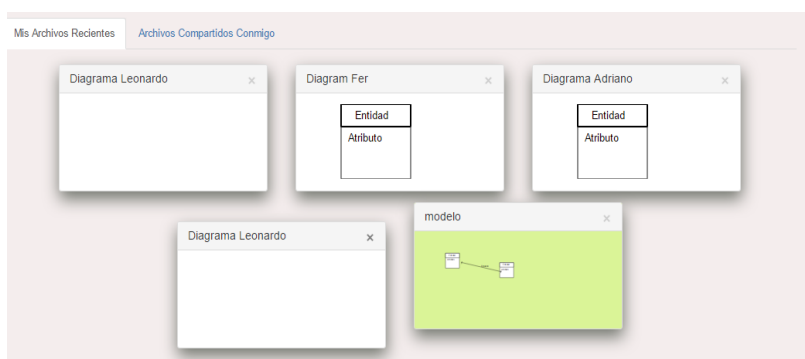
Name  
modelo

Image

```
<?xml version="1.0" encoding="iso-8859-1"?><svg version="1.1"
xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
viewBox="0 0 1000 500" style="enable-background:new 0 0
1000 500;" xml:space="preserve"><svg id="svg" class="work-
area geBackgroundPage" version="1.1"
xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">
<defs>
<filter id="shadowBlack" x="0" y="0" width="100%"
height="100%">
<feOffset result="offOut" in="SourceAlpha" dx="20"
```

Cerrar Guardar

GUARDAMOS EL  
DIAGRAMA



VISUALIZAMOS EL  
DIAGRAMA  
GUARDADO

## Script de la base de Datos en Ruby

---

```
create_table "diagrams", force: :cascade do |t|
  t.string "name", limit: 255
  t.text "image", limit: 65535
  t.datetime "created_at", null: false
  t.datetime "updated_at", null: false
end

create_table "diagrams_users", id: false, force: :cascade do |t|
  t.integer "diagram_id", limit: 4, null: false
  t.integer "user_id", limit: 4, null: false
  t.boolean "shared"
end

create_table "identities", force: :cascade do |t|
  t.integer "user_id", limit: 4
  t.string "provider", limit: 255
  t.string "uid", limit: 255
  t.datetime "created_at", null: false
  t.datetime "updated_at", null: false
end

add_index "identities", ["user_id"], name:
"index_identities_on_user_id", using: :btree

create_table "users", force: :cascade do |t|
  t.string "name", limit: 255
  t.string "facebook", limit: 255
  t.string "twitter", limit: 255
  t.string "google", limit: 255
  t.string "espol", limit: 255
  t.datetime "created_at", null: false
  t.datetime "updated_at", null: false
  t.datetime "remember_created_at"
end

add_foreign_key "identities", "users"
end
```

## *Despliegue de la aplicación*

---

### RUBY ON RAILS

1. Visitar el sitio Railsinstaller.org y elegir una de las opciones, en nuestro caso el de la derecha RailsInstaller.

## Windows Downloads



**WINDOWS**  
RUBY 2.0

Packages included are

**Ruby** 2.0.0

**Rails** 4.1

**Bundler**

**Git**

**Sqlite**

**TinyTDS**

**SQL Server Support**

**DevKit**



**WINDOWS**  
RUBY 2.1

Packages included are

**Ruby** 2.1.5

**Rails** 4.1

**Bundler**

**Git**

**Sqlite**

**TinyTDS**

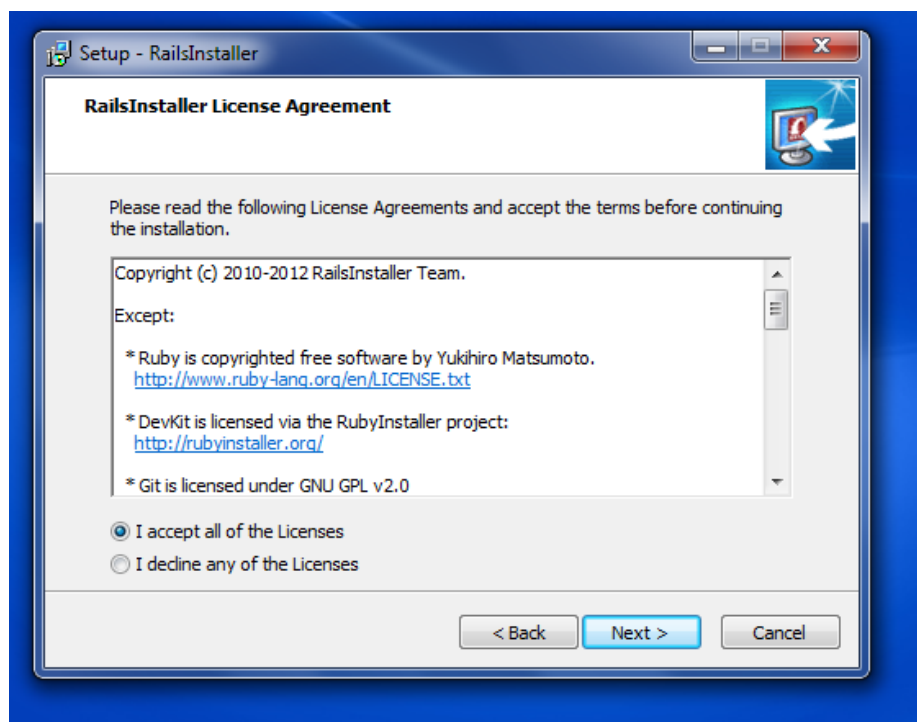
**SQL Server Support**

**DevKit**

2. Una vez, descargado, hacer doble-click para iniciar la descarga.



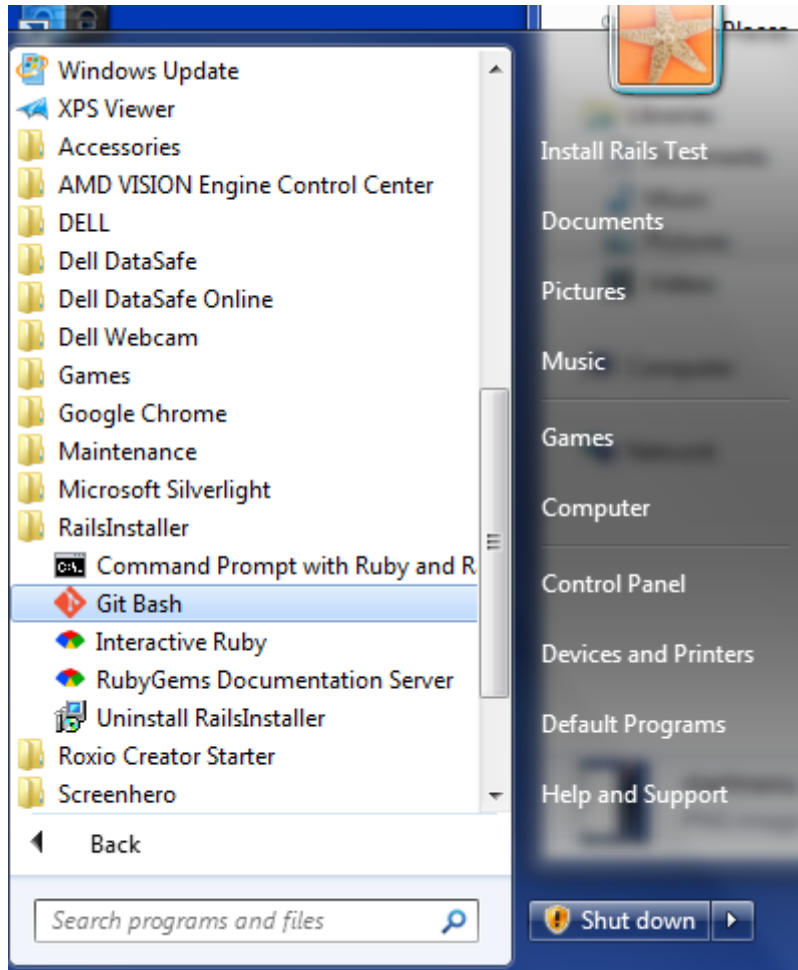
3. Colocar Accept en los default settings:



4. Instalar sublimeText(OPCIONAL).

## COMMAND LINE

5. Abrimos Git Bash desde la carpeta de RailsInstaller



6. En el prompt colocar el siguiente comando y pulsar ENTER.

```
curl http://installrails.com/update_rubygems.rb | ruby
```

## CREATE YOUR FIRST APP

7. Correr con cd en el bash hacia la carpeta donde se encuentra el Desktop por ejemplo:

```
$ cd ~/Desktop
```

8. Ahora correremos el siguiente comando para crear la aplicación:

```
$ rails new sample_app
```

9. Ingresamos a la aplicación con cd:

```
$ cd ~/Desktop/sample_app
```

10. Con esto podremos iniciar el servidor:

```
$ rails server
```

11. El comando debería mostrar algo parecido a esto si está correctamente inicializado:

```
=> Booting WEBrick

=> Rails 4.0.0 application starting in development on http://
/0.0.0.0:3000

=> Run `rails server -h` for more startup options

=> Ctrl-C to shutdown server

[2013-08-01 16:37:03] INFO    WEBrick 1.3.1

[2013-08-01 16:37:03] INFO    ruby 2.0.0 (2013-05-14) [x86_64-
darwin12.4.0]

[2013-08-01 16:37:03] INFO    WEBrick::HTTPServer#start: pid=1
2109 port=3000
```

12. Finalmente abrimos el browser y direccionamos a <http://localhost:3000> y listo.

De esa forma podemos iniciar una aplicación en Rails.

13. Como paso adicional podemos instalar un IDE, nuestra recomendación es JET  
BRAINS RUBYMINE 7.1.4

<https://www.jetbrains.com/ruby/download/#section=windows-version>



14. OJO.- Al trabajar en Windows, se cambia la configuración del application.html.erb a `default` como se aprecia en la imagen:



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Example</title>
5   <%= stylesheet_link_tag 'default', media: 'all', 'data-turbolinks-track' => true %>
6   <%= javascript_include_tag 'default', 'data-turbolinks-track' => true %>
7   <%= csrf_meta_tags %>
8 </head>
9 <body>
10
11 <%= flash.each do |key, value| %>
12   <div class="alert alert-<%= key %> alert-dismissible">
13     <button type="button" class="close" data-dismiss="alert" aria-hidden="true"></button>
14     <%= value %>
15   </div>
16 <%= end %>
17
18 <%= if current_user %>
19   Welcome <%= current_user.email %> <%= link_to 'Logout', destroy_user_session_path, method: :delete %>
20 <%= else %>
21   <%= link_to 'Login', new_user_session_path %>
22 <%= end %>
23
24
25
26 <%= yield %>
27
28 </body>
29 </html>
30
```

15. Para correr la aplicación desde el RubyMine, ir a la opción Abrir(Open), y encontrar la ruta del proyecto, en nuestro caso es ../uDraw y listo.