

Actividad Presencial Semana 15

Autenticación con Devise (Experiencia presencial 2)

Para poder realizar esta actividad debes haber realizado los cursos previos junto con los videos online correspondientes a la experiencia 15.

El objetivo de esta actividad es la implementación de un Sistema de autenticación con Devise para permitir el ingreso de un usuario en una aplicación.

Setup:

En esta actividad se utilizó la gema carrierwave para el manejo de archivos por lo que antes de empezar tendrás que instalar imagemagik en tu computador. Si ya lo has instalado antes, omite este paso.

Para saber si ya está instalado vamos a escribir:

```
convert -version
```

Si obtenemos un mensaje de command not found entonces debemos instalar la herramienta.

Instalando ImageMagick

En **OSX**:

```
brew update
```

```
brew install imagemagick
```

EN **Ubuntu**:

```
sudo apt-get update  
sudo apt-get install imagemagick
```

Más información en [Documentación carrierwave](#)

Comienza la actividad

Pictory es una aplicación para que diversos usuarios guarden sus historias y puedan compartirlas, pero esta aplicación no está terminada, el cliente pide:

- Realizar un fork y descargar la aplicación.
- Agregar la gema *devise* al gemfile.

```
gem 'devise', git: 'https://github.com/plataformatec/devise.git'
```

- Generar los archivos de devise.
- Crear un modelo de **User** con **Devise**
- Crear las vistas de devise.
- Aplicar diseño a las vistas de devise, acorde al diseño de la aplicación.
- Agregar a **History** una foreign key que haga referencia al usuario.
- Al momento de crear una nueva historia, asignar el usuario creador a la historia creada.

Esto debe hacerse en el método create en controlador de Historias

- Añadir el campo name (string), username (string) y admin (boolean) en el modelo **User**.

- Añadir los campos name y username a los formularios de devise.
- Agregar los campos nuevos a los strong paramas > Revisar documentación de [devise](#).
- Validar en modelo user el campo name como obligatorio, y el campo username como obligatorio y único.
- Modificar el menú para que, cuando el usuario no se encuentre conectado, muestre los link de login y registro, y cuando se encuentre conectado, muestre los link de editar registro y cerrar sesión.
- El usuario no conectado solo podrá ver index de History, si quiere entrar en otra página debe solicitar login.
- Modificar la vista index de History, si el usuario no está conectado solo mostrará el botón de show en cada uno de los thumbnails.

No se mostrarán el editar y destruir.

- Si el usuario está conectado, el usuario solo podrá modificar las historias que le pertenecen.
- Si el usuario conectado es admin, el usuario podrá modificar todas las historias.
- Crear vista con las historias que le pertenecen al usuario.
- Subir la aplicación funcionando a Herokul

Parte avanzada

- Crear un panel de control de usuarios al que solo tendrán acceso los usuarios admin.

El panel de control es solo una acción especial nueva, que muestra todos los usuarios Esta acción solo debe ser accesible para un usuario con el rol admin.

- Dentro del panel de control de usuarios, añadir al formulario de user la opción para dar o quitar el privilegio de admin.
 - Estas acciones solo deben estar disponibles para un usuario con el rol admin.