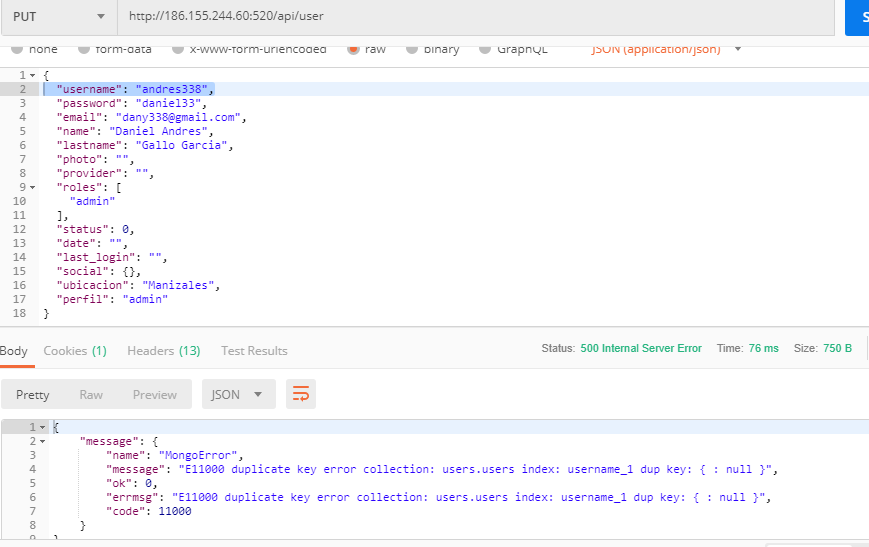
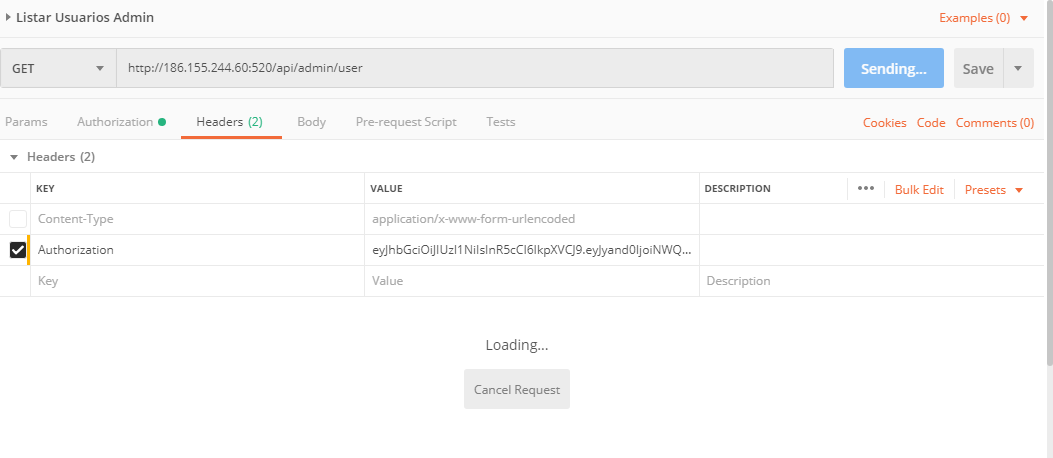
El api para el actualizar no funciona correctamente



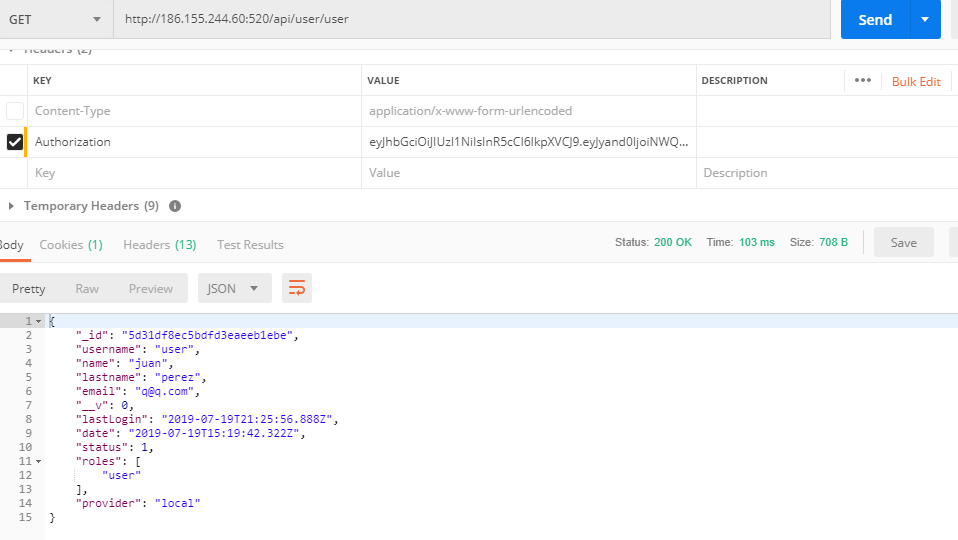
El api para listar los usuarios tampoco funciona, y solo esta diseñada para el rol admin

/api/admin/user Get the list of users [Role admin]

Cuando los usuarios se crean se crean por defecto con el rol user, la petición se queda enviando…



Funciona la ruta <http://186.155.244.60:520/api/user/user>, pero igual no nos da la información de los usuarios creados con el rol user por defecto, por lo que se simulo en un json el listado de usuarios.



El manejo de estados y propiedades de la aplicación se manejo utilizado redux la única fuente de la verdad.

**Definición de componentes**

App: En el componente ha se encarga de inicializar el provider con el store, el estado es manejado con redux y con react router se definen las rutas que tendrá la aplicación.

CardHome: Se encarga de renderizar cada usuario en una tarjeta con su información correspondiente.

CardLogin: Se encarga de renderizar el login para ingresar a la app.

Filters: Se encarga de renderizar en un componente los filtros de búsqueda sobre el listado de usuarios.

Header: Se encarga de renderizar el formulario que contiene los datos básicos de: nombre de usuario, nombres, apellidos, correo electrónico,

Nav: El componente nav se encarga de mostrar un logo alusivo al app, y al lado derecho el nombre de usuario indicando que se encuentra logueado y un botón de logout.

**Definición de páginas**

Home: Muestra los usuarios renderizados en tarjetas, con sus datos correspondientes.

Login: Muestra el login renderizado en una tarjeta.

From: Muestra la página de formulario para crear nuevos usuarios.

**Definición de las acciones:**

login: Define las acciones que se harán sobre parte del estado en este caso login

isLogin, errorLogin

users: Define las acciones que se harán sobre parte del estado en este caso users

, loading, error

filters: Define las acciones que se harán sobre parte del estado en este caso selectedFilter, filters

index: exporta todas las acciones

**Definición de los reducers:**

filters: Especifica el cómo hacer las acciones sobre ciertas partes del estado en base a los tipos de acciones en este caso para manipular el objeto selectedFilter y filters.

login: Especifica el cómo hacer las acciones sobre ciertas partes del estado en base a los tipos de acciones en este caso para manipular el objeto login.

users: Especifica el cómo hacer las acciones sobre ciertas partes del estado en base a los tipos de acciones en este caso para manipular el array de objetos users.

index: combina todos los reducers

**Definición del estado general para toda la aplicación:**

index: Definición del store con sus dispatchers y middlewares para lo de las peticiones al servidor.

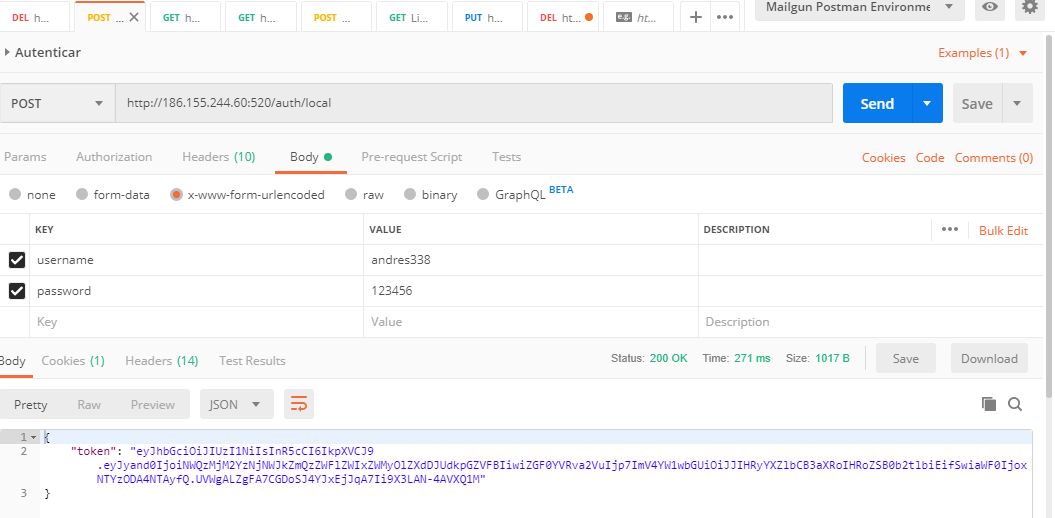
initialState: Definición de como esta conformado el estado general de la aplicación.

dispatchers: Definición de que acciones se envían – disparan al estado

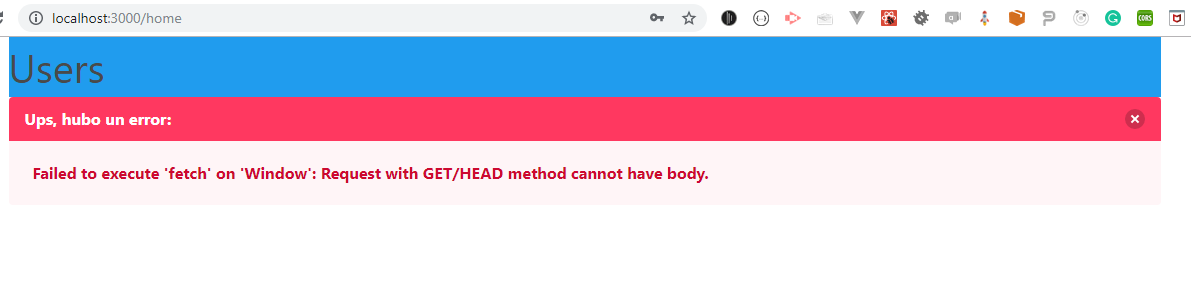
**Definicición de utilidades:**

api: Definición de los request – peticiones a mandar – solicitudes api al servidor.

Funcionamiento:



Cuando hay un error en las peticiones de la app saca un mensaje de error:

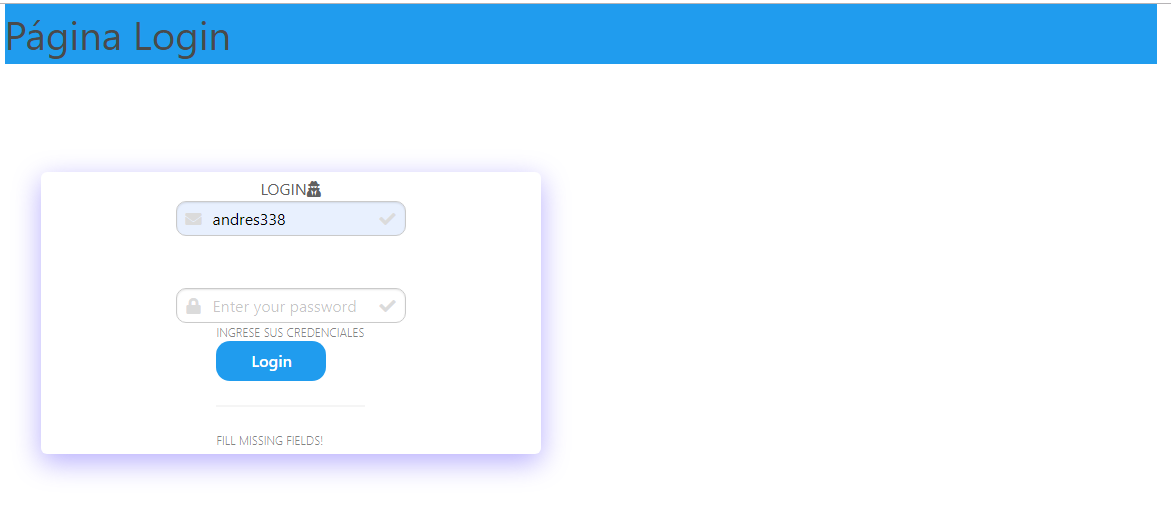


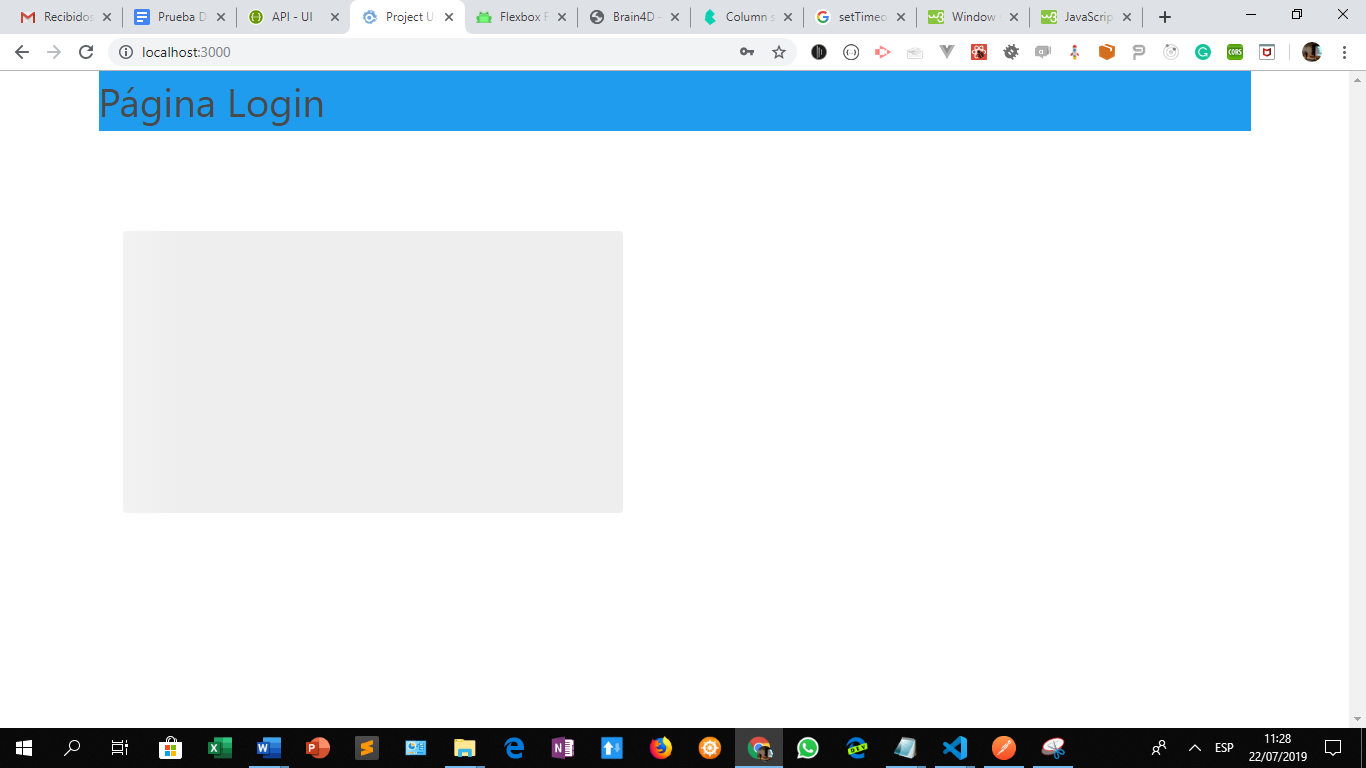
Las credenciales de acceso para el logueo son:

Username: andres338

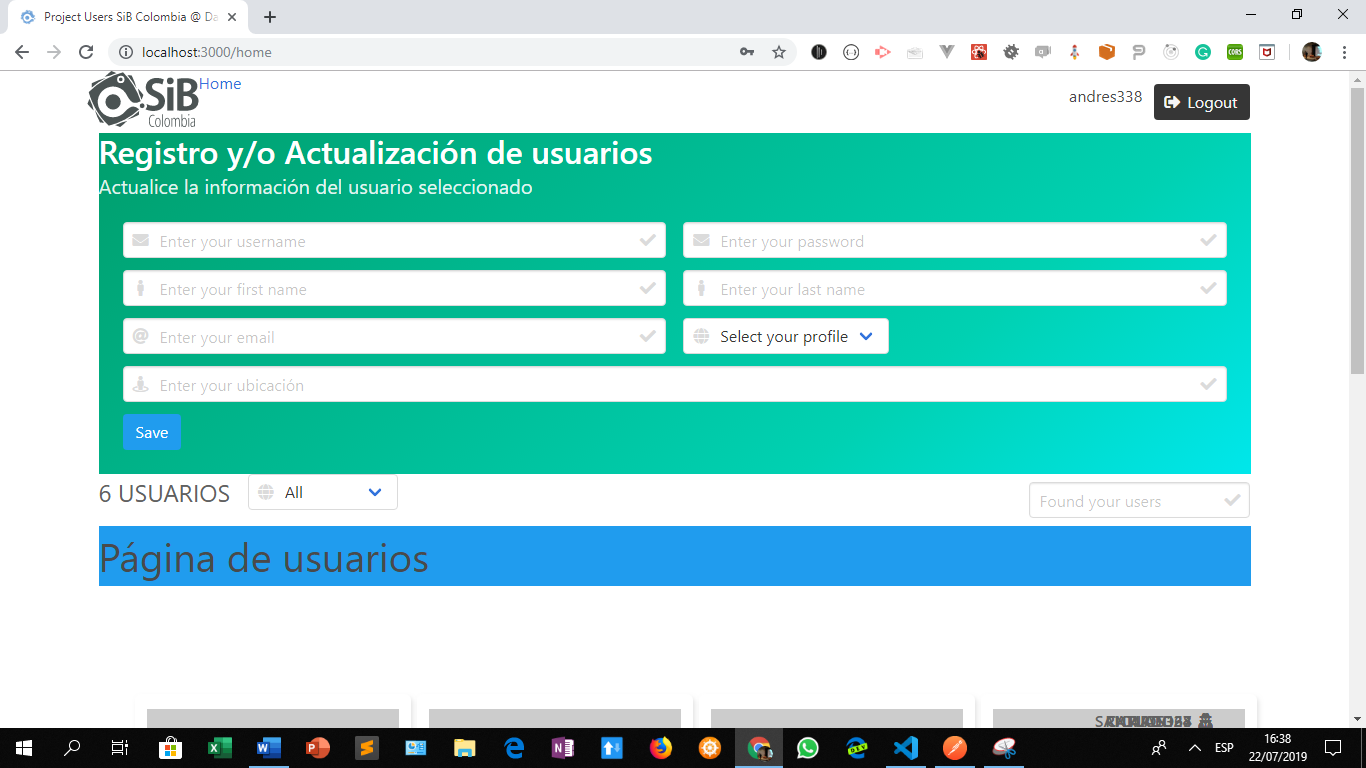
Password: 123456

Se valida que todos los datos hayan sido ingresados:

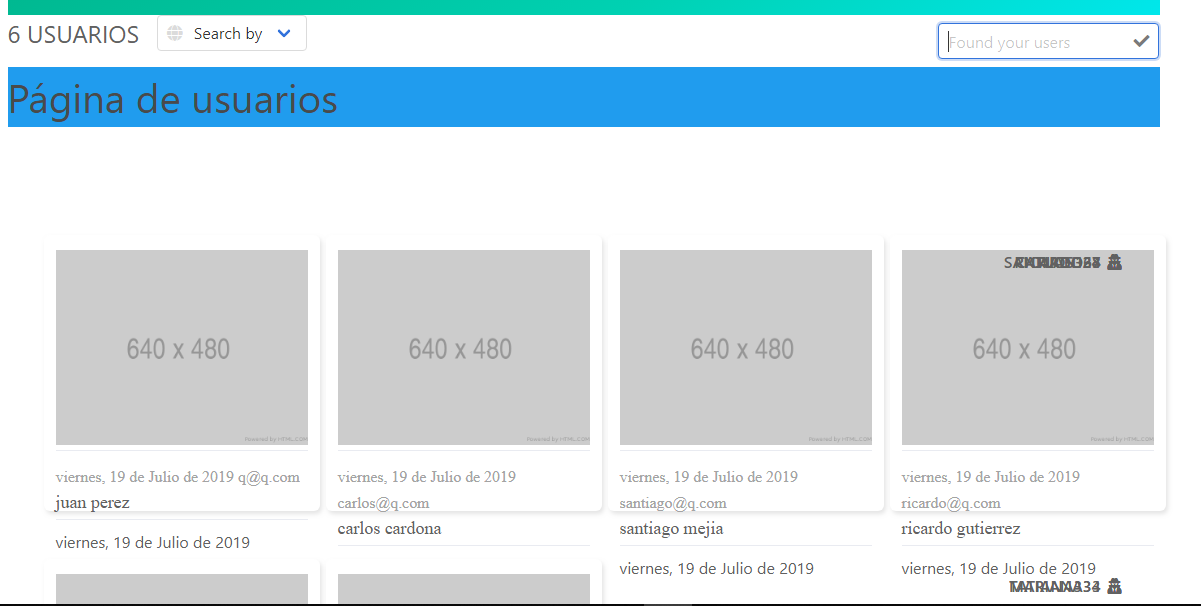




Cuando se loguea correctamente aparece un formulario para ingresar los datos correspondientes

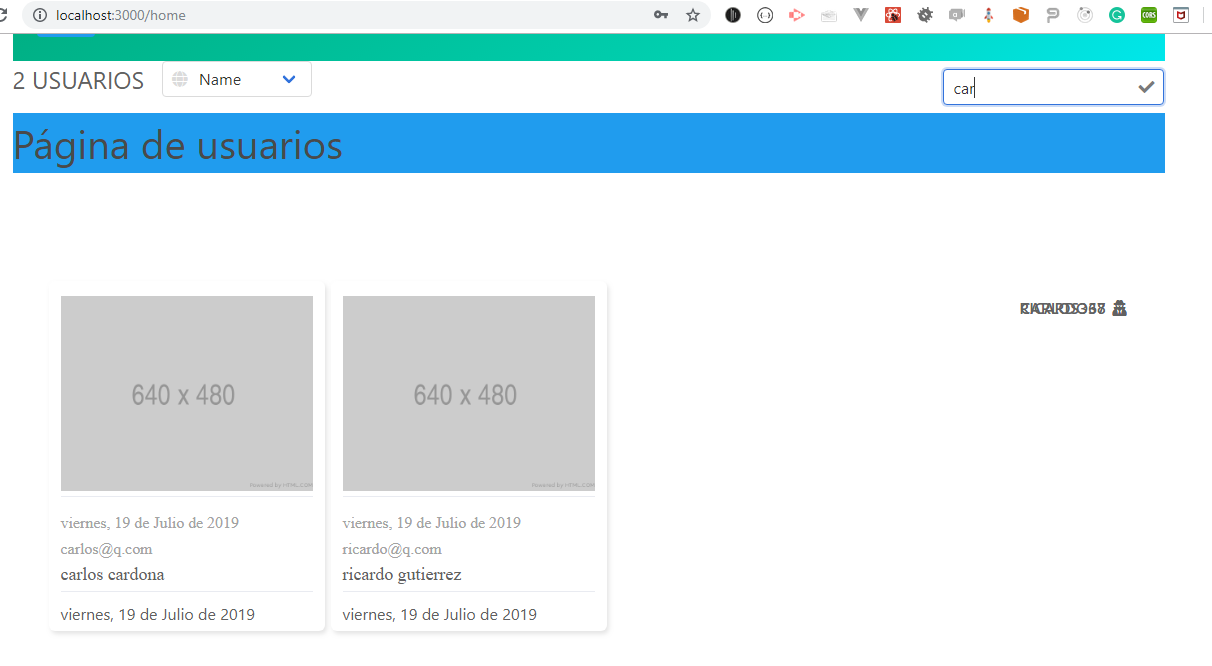


Aprece el filtro con el listado correspondiente

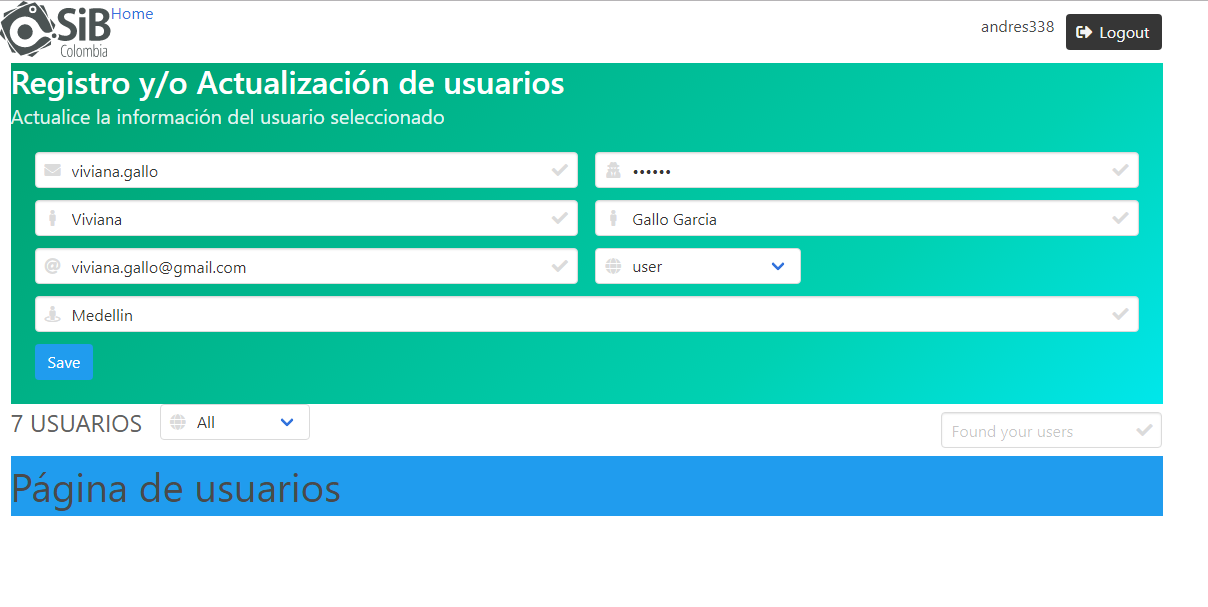


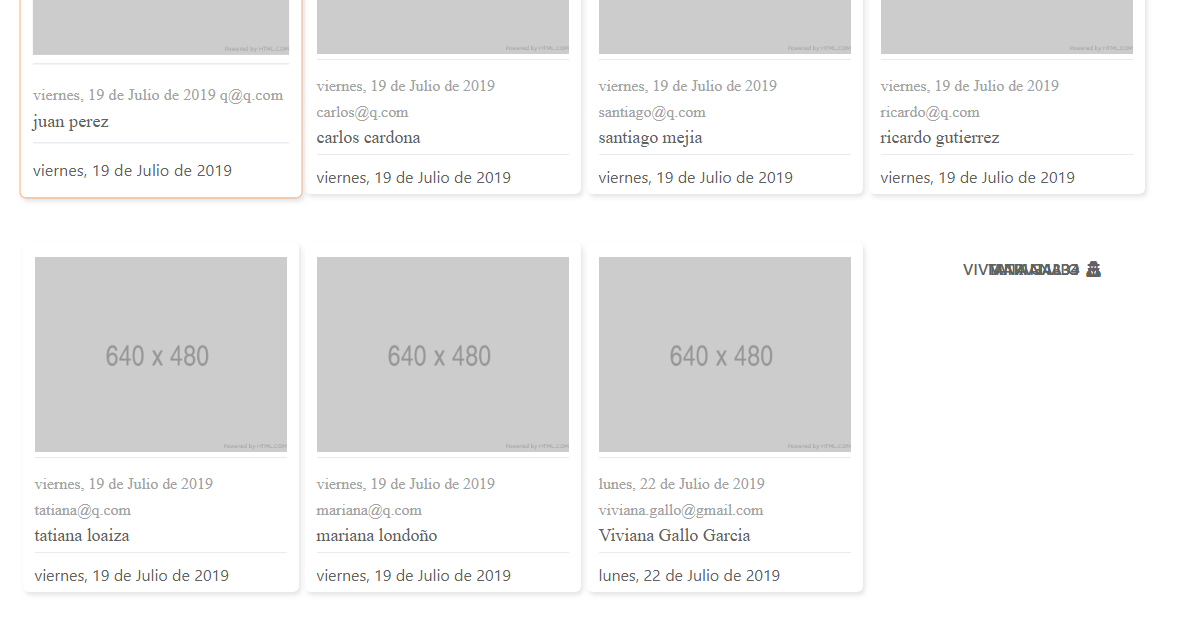
El diseño de los cards se usó styled components

Cuando se realiza la búsqueda aparece de la siguiente forma:



Cuando se crea un nuevo usuario:





Cuando las credenciales son incorrectas:

