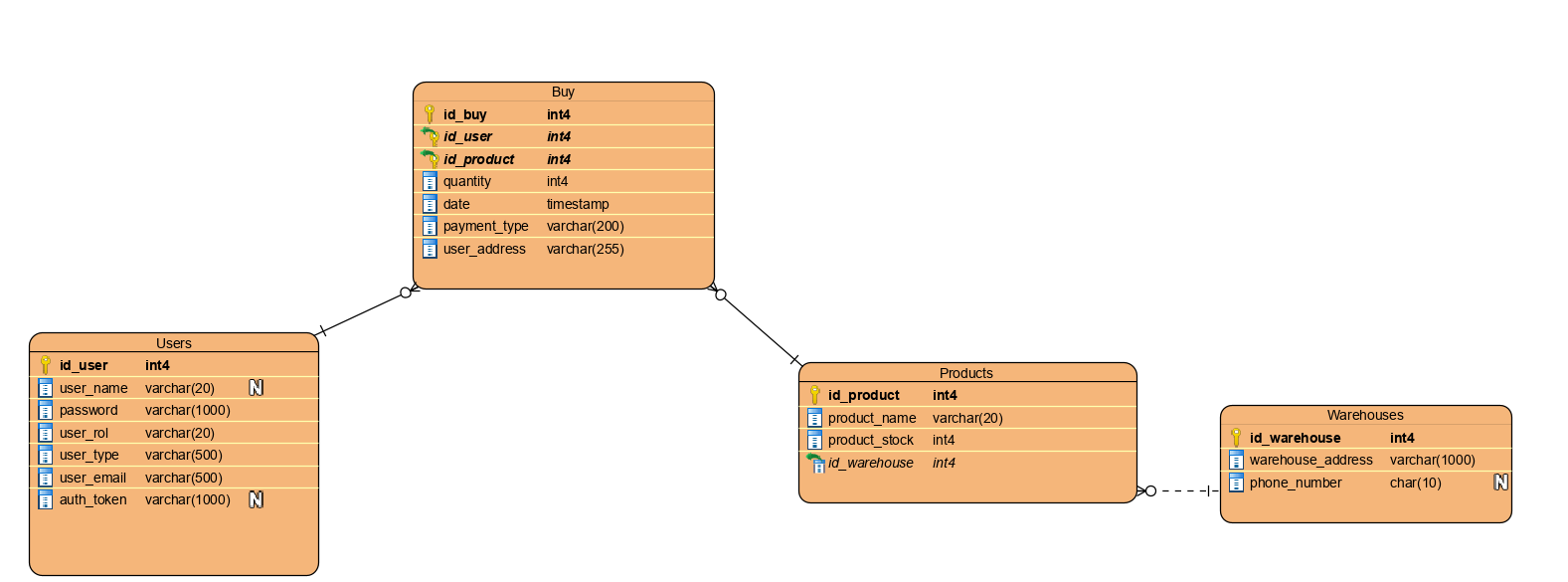
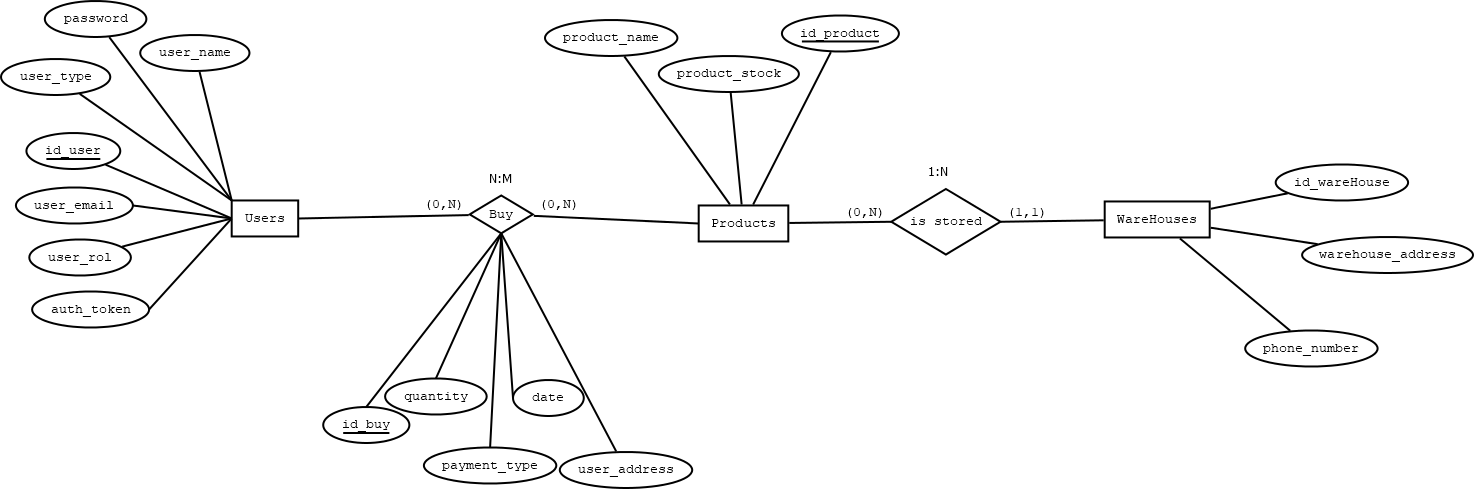
**FullStack Application**

**Descripción de la aplicación**

* Pongamos que yo tengo un bar y yo quiero expandir y vender de manera online eso hará mi aplicación y claro al hacer eso ocurre una necesidad saber donde se encuentran nuestros productos por eso creamos un almacen para llevar un registro
* Diagrama ERD:



* Diagrama E-R:



* Modelo relacional:

Users(id\_user,user\_name,password,

user\_rol,user\_type,user\_email,image\_profil,auth\_token)

Buys(id\_buy,id\_user\*,id\_product\*,quantity,

date,payment\_type,user\_address)

Products(id\_product,product\_image,id\_warehouse\*,

product\_name,product\_stock)

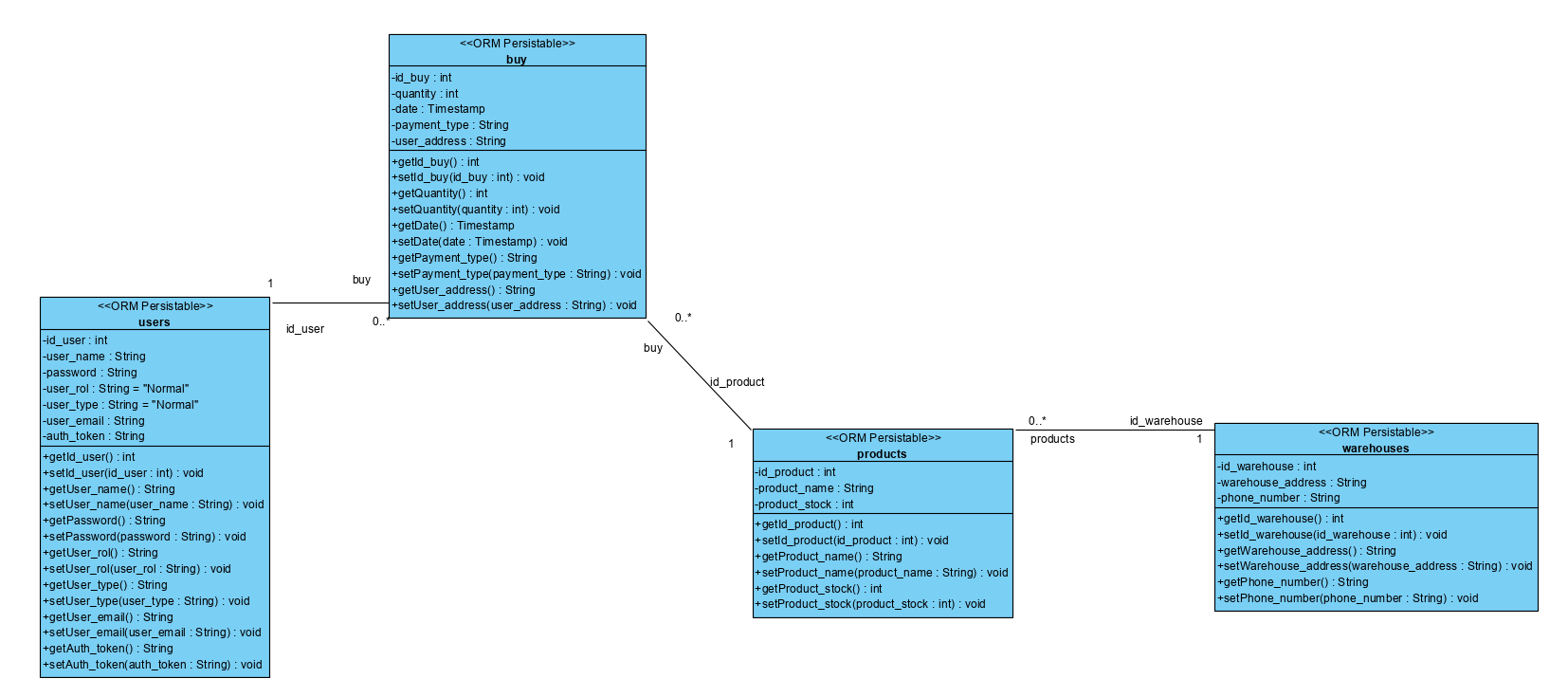
Warehouses(id\_warehouse,warehouse\_address,phone\_number)

* Pila tecnológica:





**Modelo de datos**



# **Descripción de cada tabla/atributo**

* **Users:**

La función de esta tabla es almacenar todos los datos de los usuarios junto con un token para la autentificación.

**-id\_user:** Identificador del usuario.

**-user\_name:** Es el nombre del usuario.

**-user\_rol:** Se utiliza para saber si el usuario es administrador.

**-user\_type:** Se utiliza para saber qué tipo de cuenta es [“Normal”,” Google”, “Facebook”].

**-user\_email:** Email del usuario.

**-profile\_image:**Imagen del usuario

**-auth\_token:** Token de conexión del usuario.

* **Buy:**

Esta es la tabla de relación entre products y users esta sirve para crear una compra de productos con sus datos correspondientes.

**-id\_buy:** Identificador de la compra.

**-id\_user:** Identificador del usuario que ha hecho la compra.

**-id\_product:** Identificador de los productos pedidos.

**-quantity:** Cantidad de cada producto.

**-date:** Fecha del pedido.

**-payment\_type:** Método de pago [“tarjeta”, “efectivo”].

**-user\_address:** Dirección del cliente.

* **Products:**

Esta es la tabla que almacena los datos de los productos existentes y de quien los suministra.

**-id\_product:** Identificador del producto.

**-product\_name:** Nombre del producto.

**-product\_stock:** Cantidad del producto que hay en stock.

**-product\_image:**Imagen del producto

**-id\_provider:** Proveedor que suministra el producto.

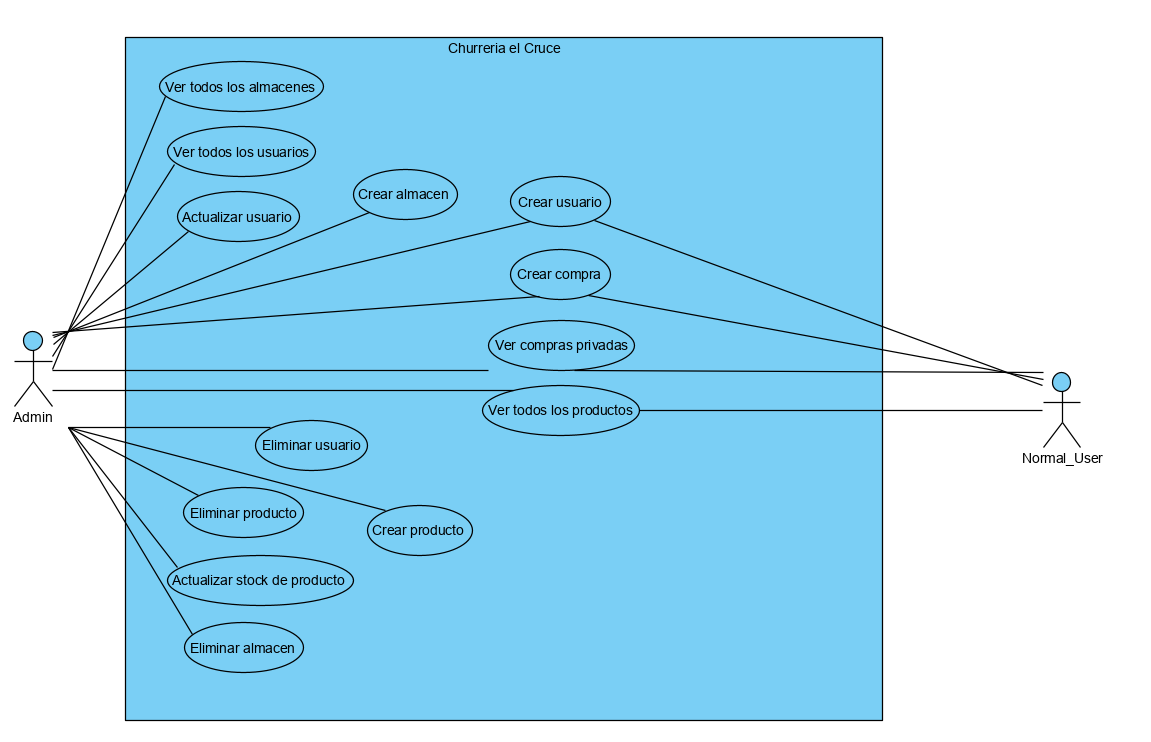
* **WareHouses:**

Esta es la tabla que almacena los datos de los almacenes.

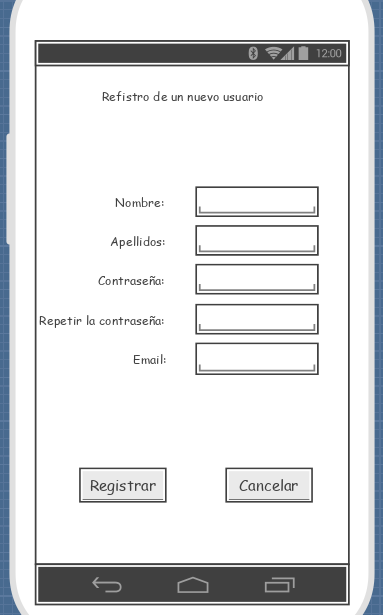
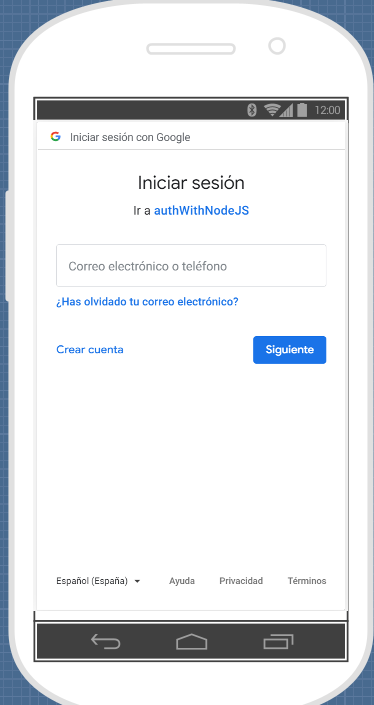
**-id\_warehouse:** Identificador del alamcen.

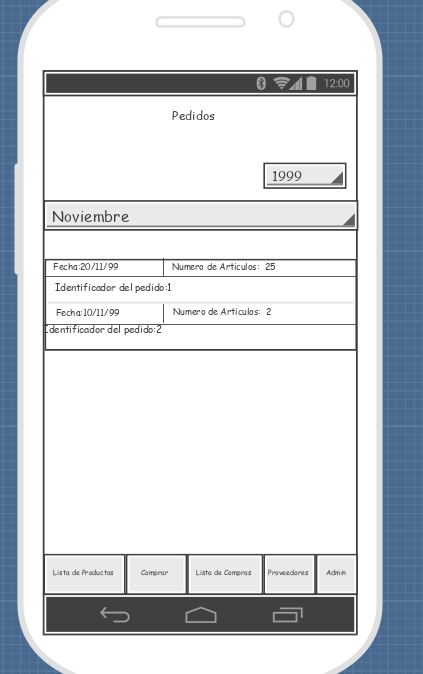
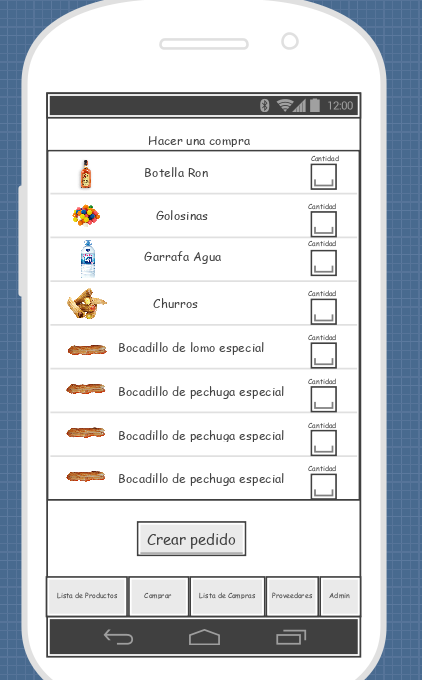
**-warehouse\_address:** Direccion del almacen.

**-phone\_number:** Telefono del almacen.

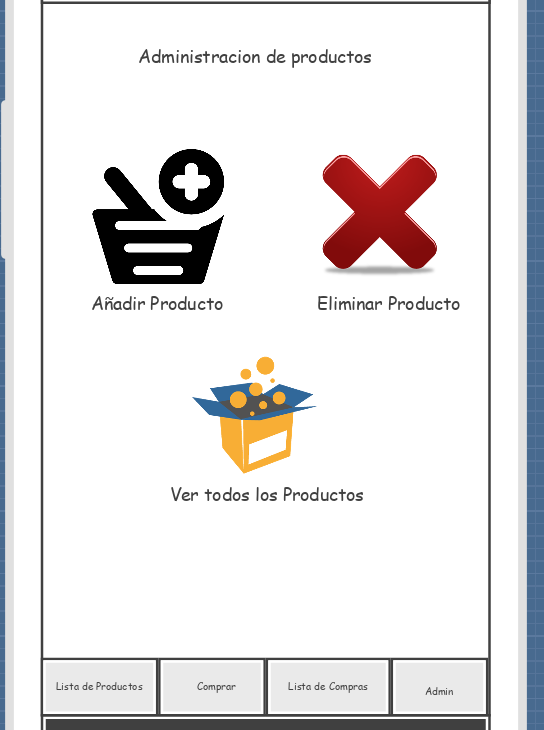
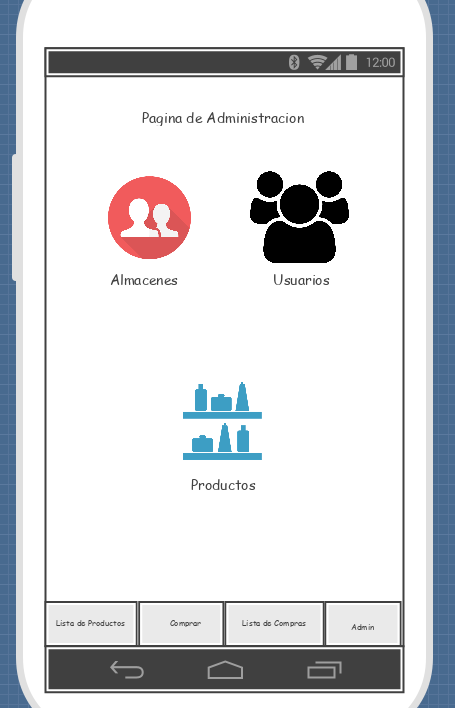
* Casos de uso

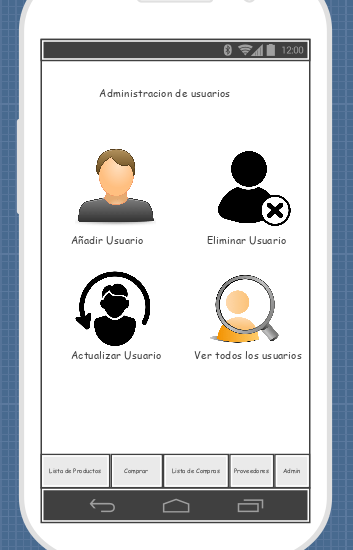
# **Pre diseño de la aplicación**



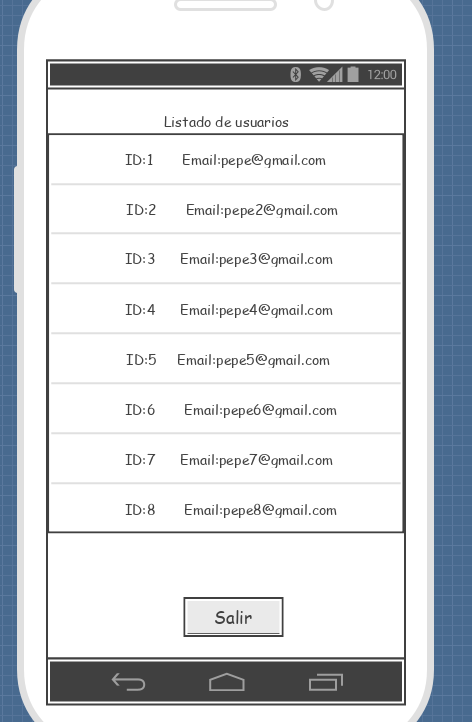
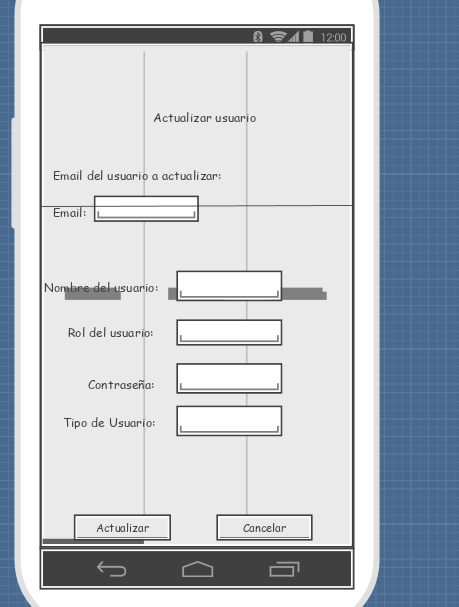
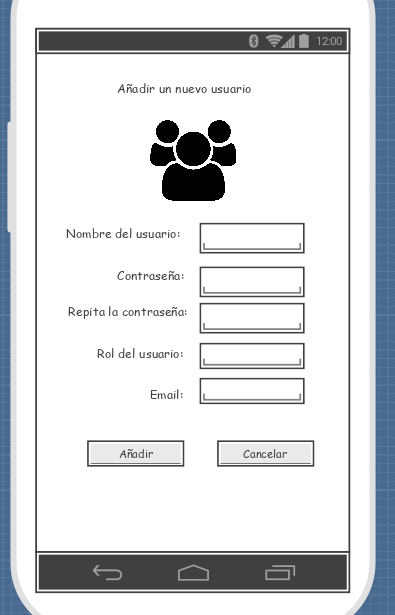




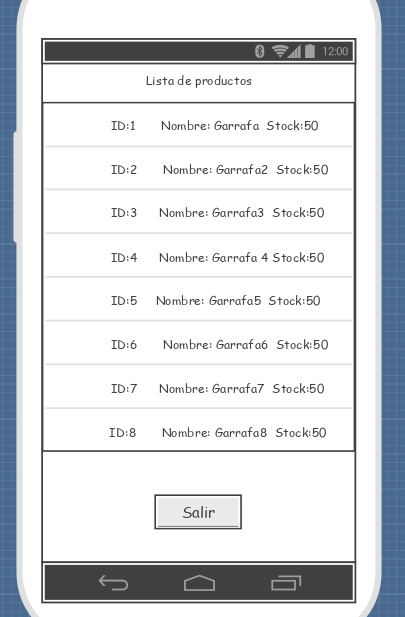
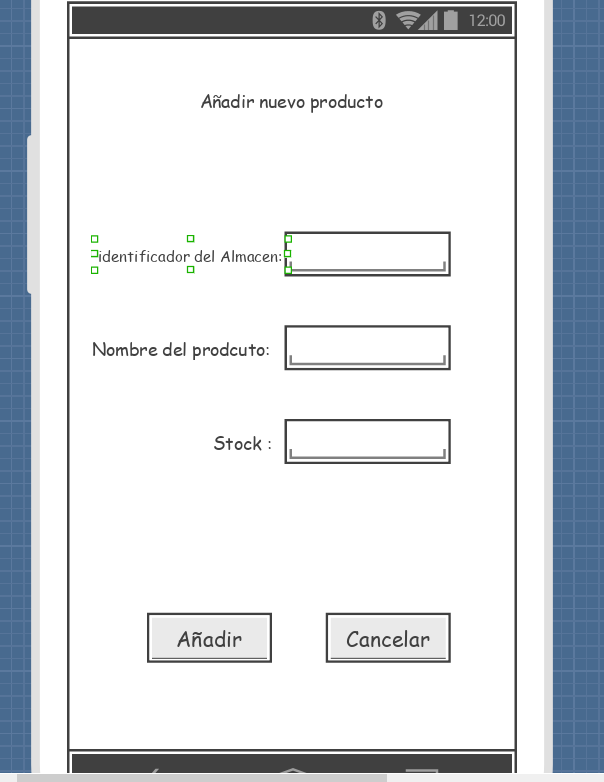




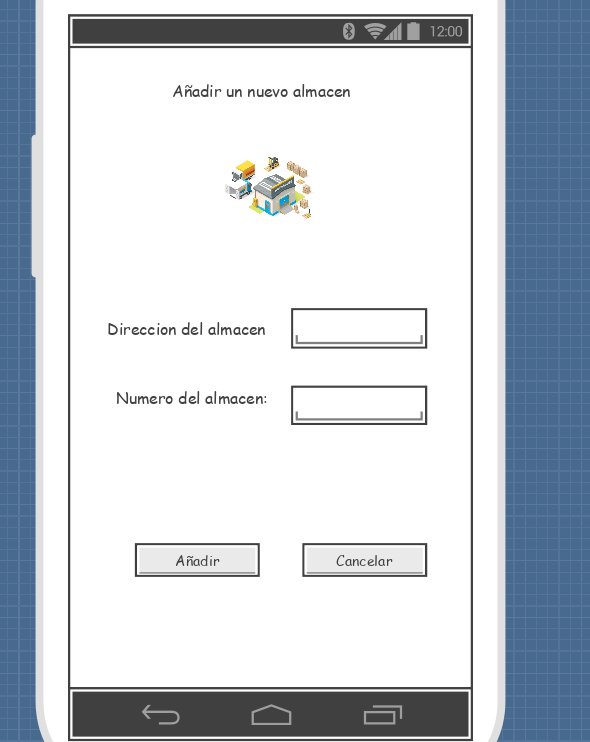
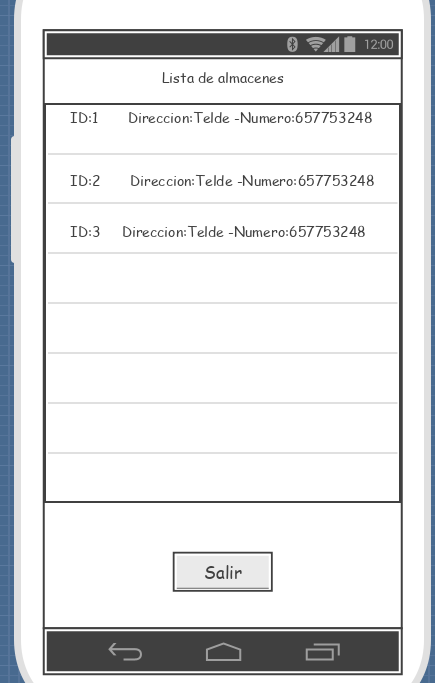
Vista del administrador para crear un usuario, actualizar usuario, listas de usuarios:



Vista del administrador para crear un producto, listas de productos tengan:



Vista del administrador para crear un almacen,eliminarlo y ver los que hay:



# Comparación de tecnologías

En estén punto compararemos Ionic con la herramienta de Android Studio que usa Android.

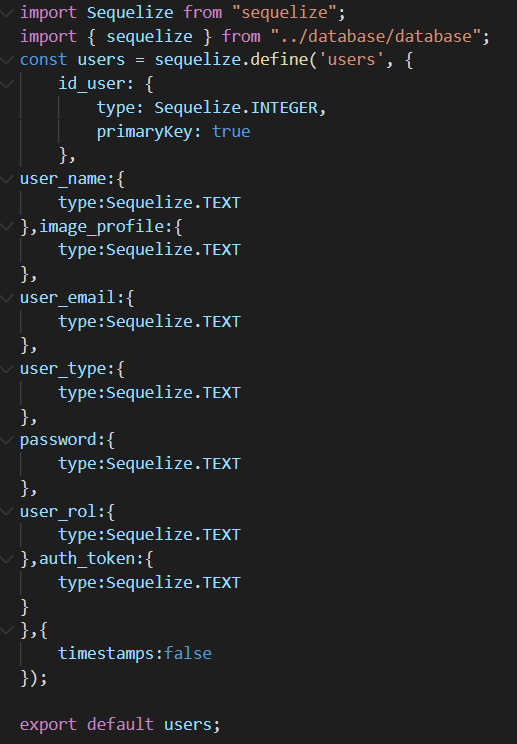
Ionic tiene ventaja principal que se programa directamente para dos plataformas que son Android y IOS. Esto facilita mucho la labor del programador además de que es muy fácil hacer una aplicación de un aspecto bonito de manera muy sencilla.

La desventaja de Ionic es que para sus pruebas son en web y cosas como la autenticación con google se han de probar en un emulador.

La desventaja de Android Studio es que consume muchos recursos y se necesita un ordenador bastante potente.

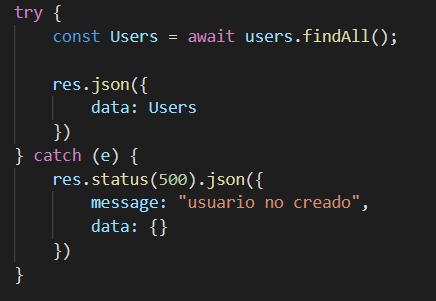
# Descripcion del mapeo ORM y consultas

Los siguiente es un modelo el cual se utiliza para mapear y tener objetos para el ORM de sequelize, esto se hace con cada una de las tablas para asi crear sus objetos.

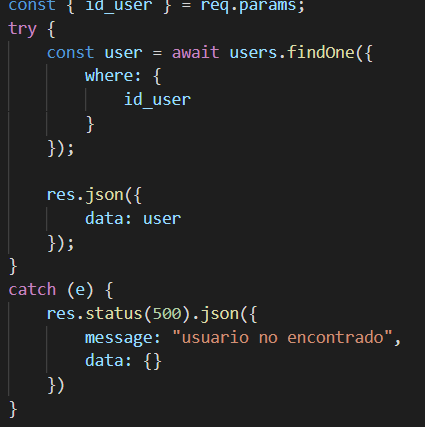


Consultas:

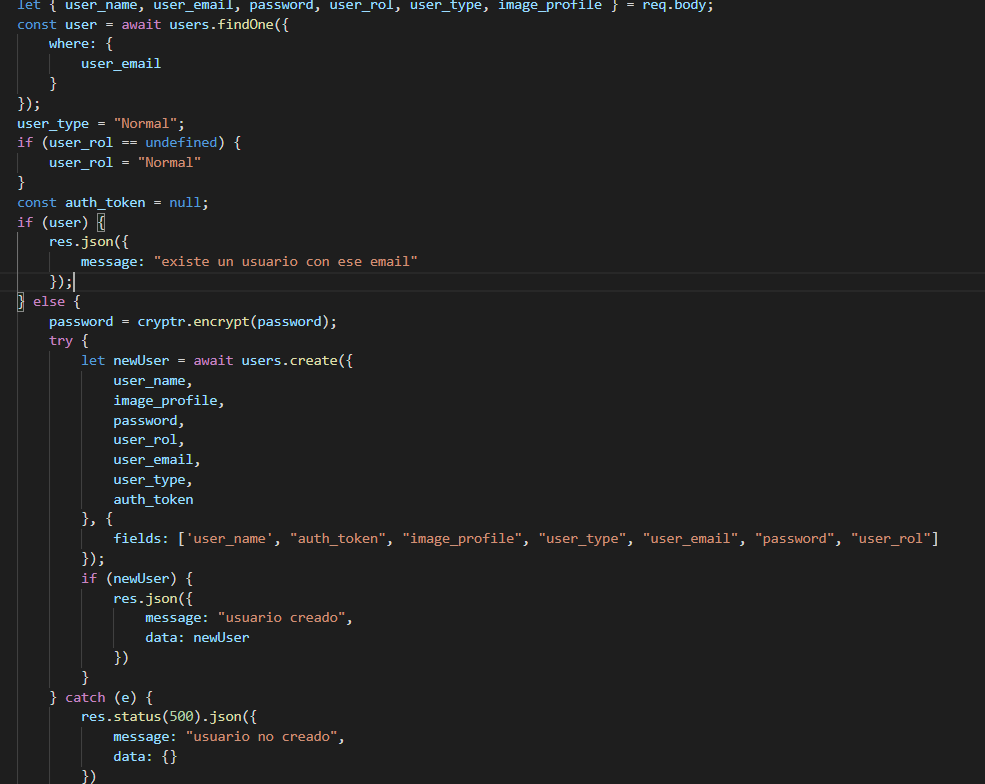
El siguiente código es para hacer un get a toda una tabla y obtener todos sus datos.



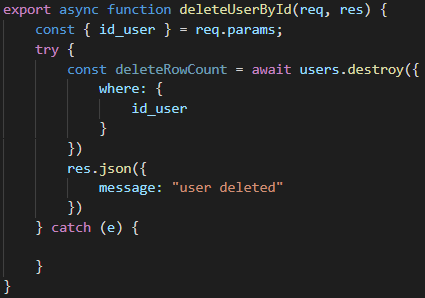
El siguiente código es para hacer un get by id para obtener los datos en concreto de un usuario en este caso.



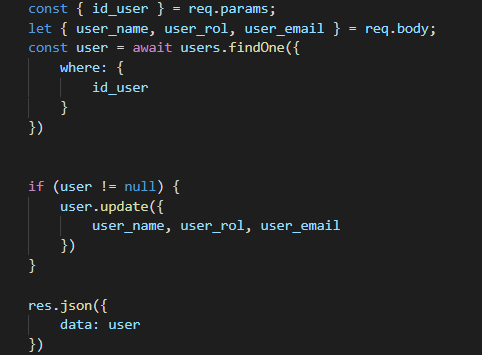
El siguiente código es un créate para en este caso crear un usuario, en este caso en este código se realiza una consulta previa para ver si existe ya un usuario con el email introducido.



En el siguiente código se eliminara un usuario.



Y en este ultimo fragmento de código se actualizara un usuario haciéndose la comprobación de que el usuario existe.



# Documentacion API

Enlace a postman: <https://documenter.getpostman.com/view/8853097/SWEB1FD8?version=latest>