|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Materia:** |  | Programación Móvil II | | |
| **Práctica:** |  | Content Provider implementado en el proyecto MiBaseDatos | | |
| **Alumno(s):** |  | Jesús García Páramo  Irving Nahum Paniagua Mora | **Fecha:** | 17/02/2020 |

**Objetivo**

Crear un proyecto que permita realizar las operaciones (CRUD) de insertar, borrar, editar y buscar en una base de datos (SQLite).

**Temas del plan de estudios**

TEMA 8: SQLite

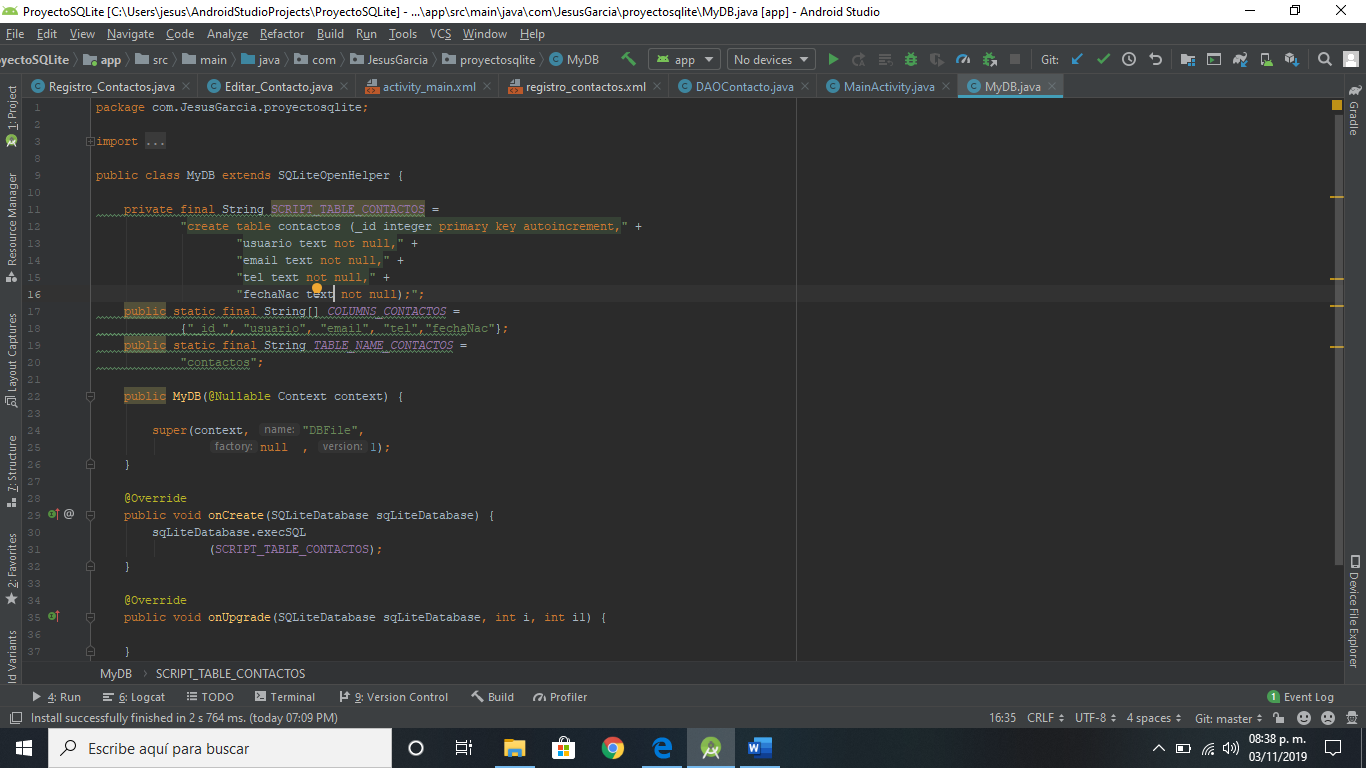
**Material**

Android Studio

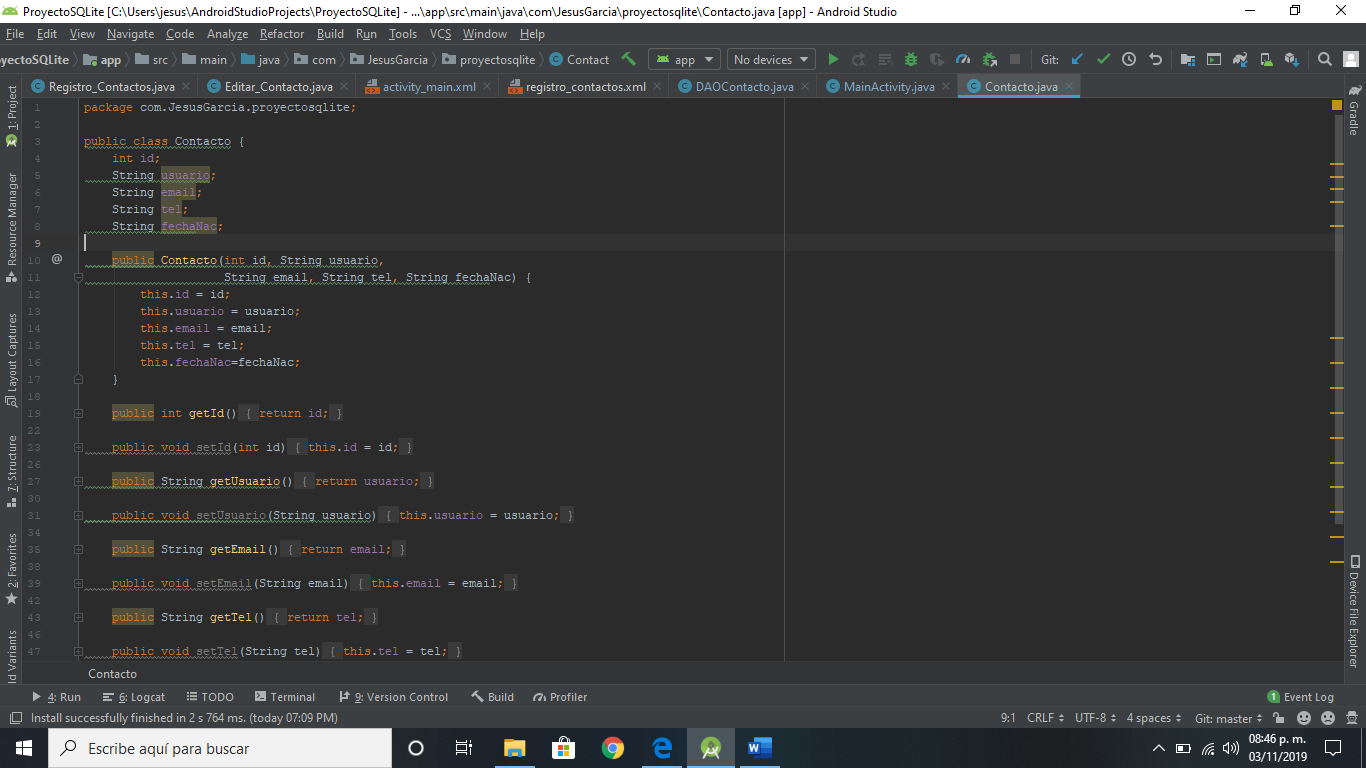
**Marco Teórico**

**Desarrollo**

**Se creo la clase MyDB que extiende de SQLiteOpenHelper la cual será la base de datos se creo la tabla contactos con las columnas: \_id de tipo integer la cual será la llave primaria y es autoincrementable, usuario, email, tel (telefono), fechaNac (fecha de nacimiento) de tipo text.**



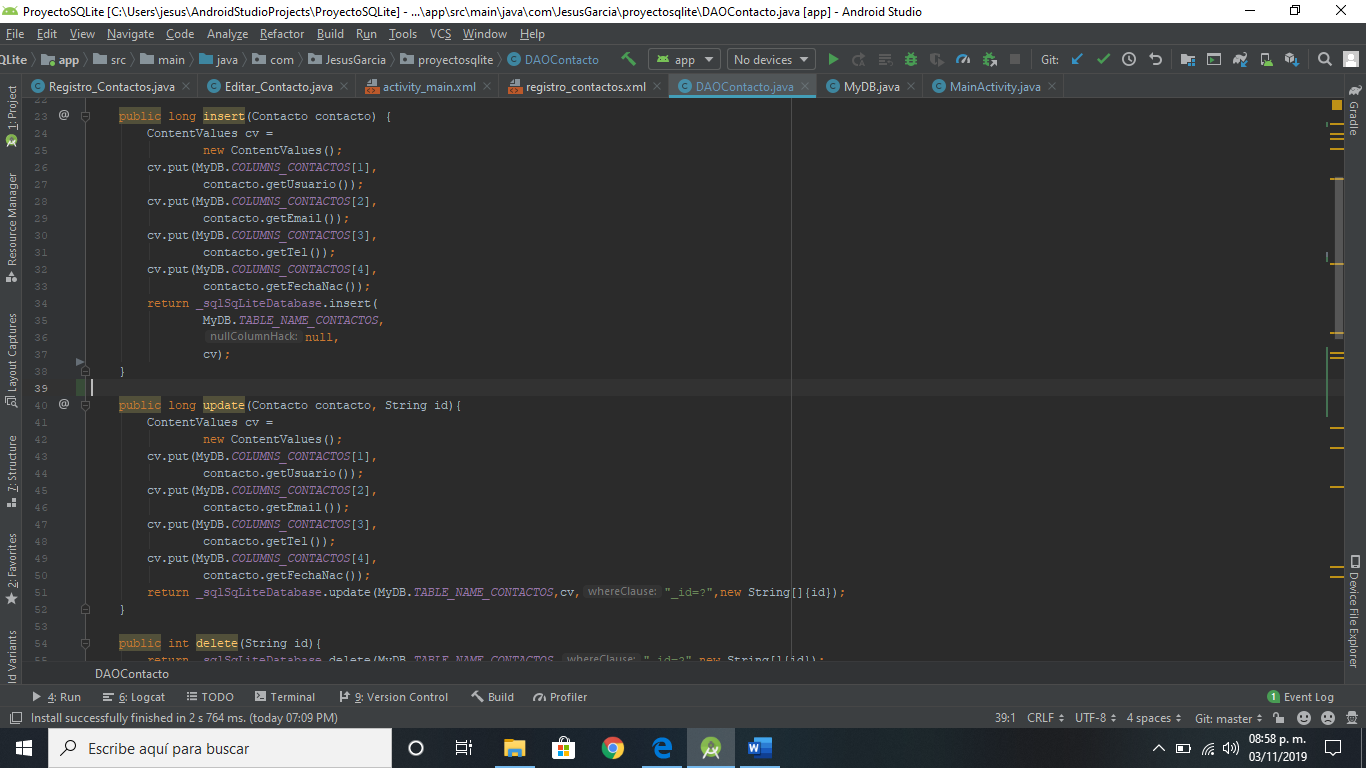
**Creamos la clase Contacto mediante la cual utilizaremos para crear los contactos mediante los atributos: id, usuario, email, tel (N° de telefono) y fechaNac (fecha de nacimiento), la clase cuenta con su respectivo constructor, getters y setters.**



**Creamos la DAOContacto la cual contendrá los métodos de insert, update, delete, filter y el getAll para obtener una lista de todos los contactos de la base de datos.**

**El método insert recibe un objeto de tipo Contacto del cual solo obtendremos el usuario, email, tel, y fechaNac y lo asignaremos a su respectiva columna de la tabla contactos en la base de datos el id no lo estamos asignado porque es de tipo auto incrementable.**

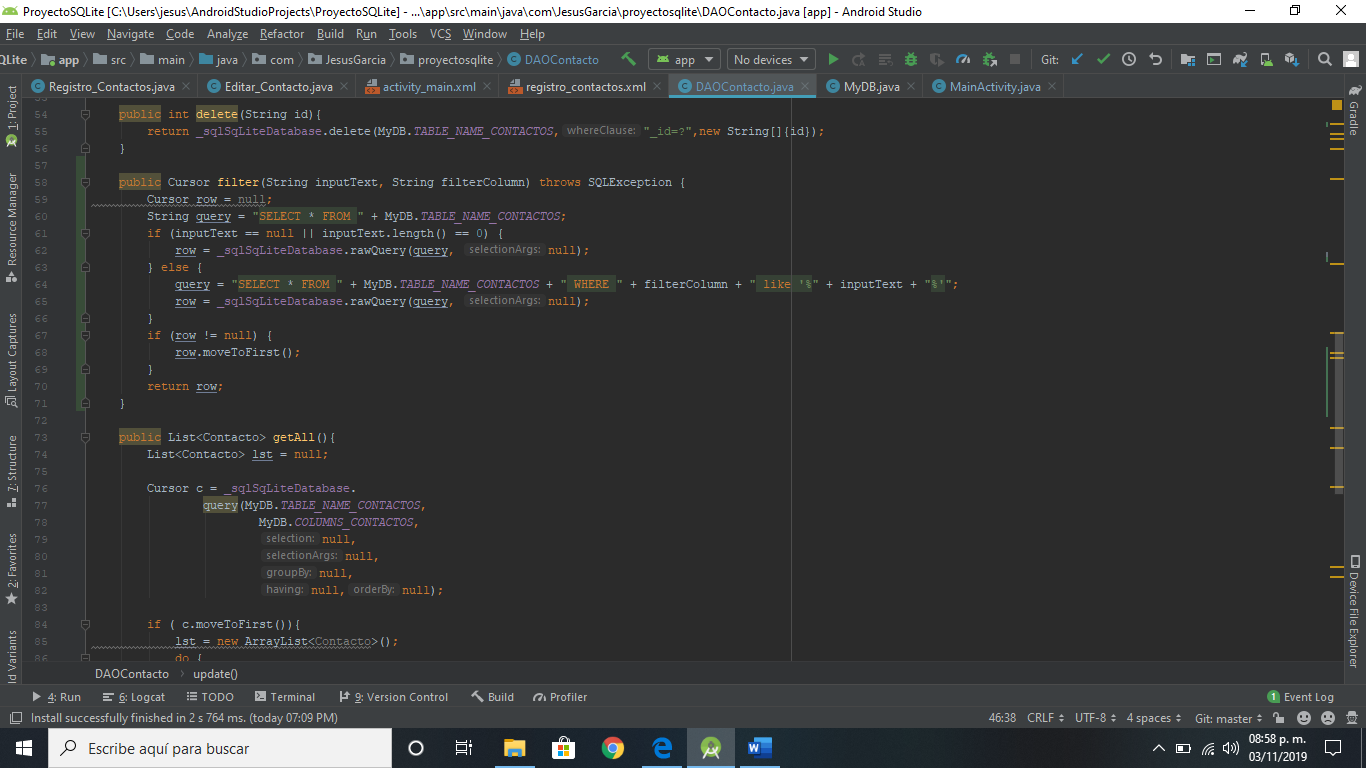
**El método update recibe un objeto de tipo Contacto del cual obtendremos los datos del contacto y un id que seria el id del usuario seleccionado donde posteriormente se hará una modificacion de ese usuario.**

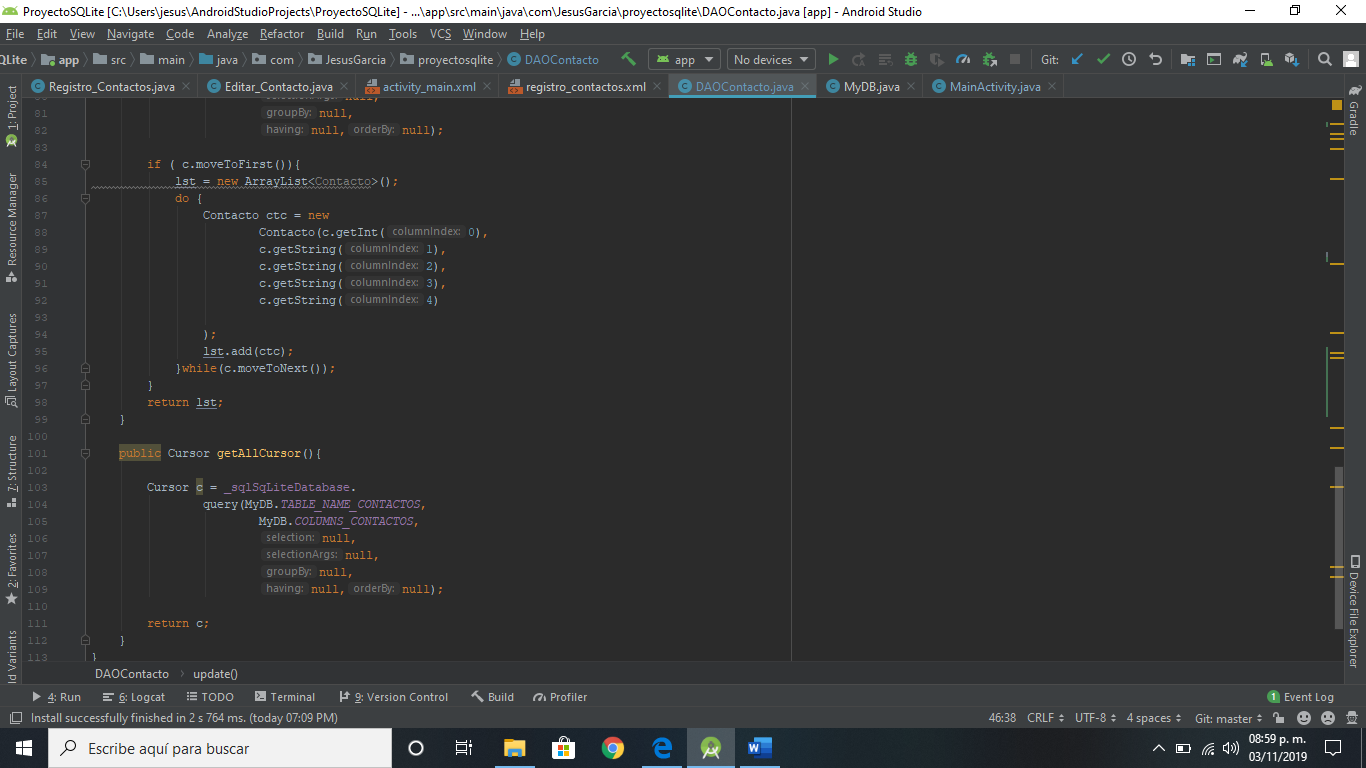


**El método delete recibe una cadena la cual será el id de usuario que se seleccionó para eliminarlo.**

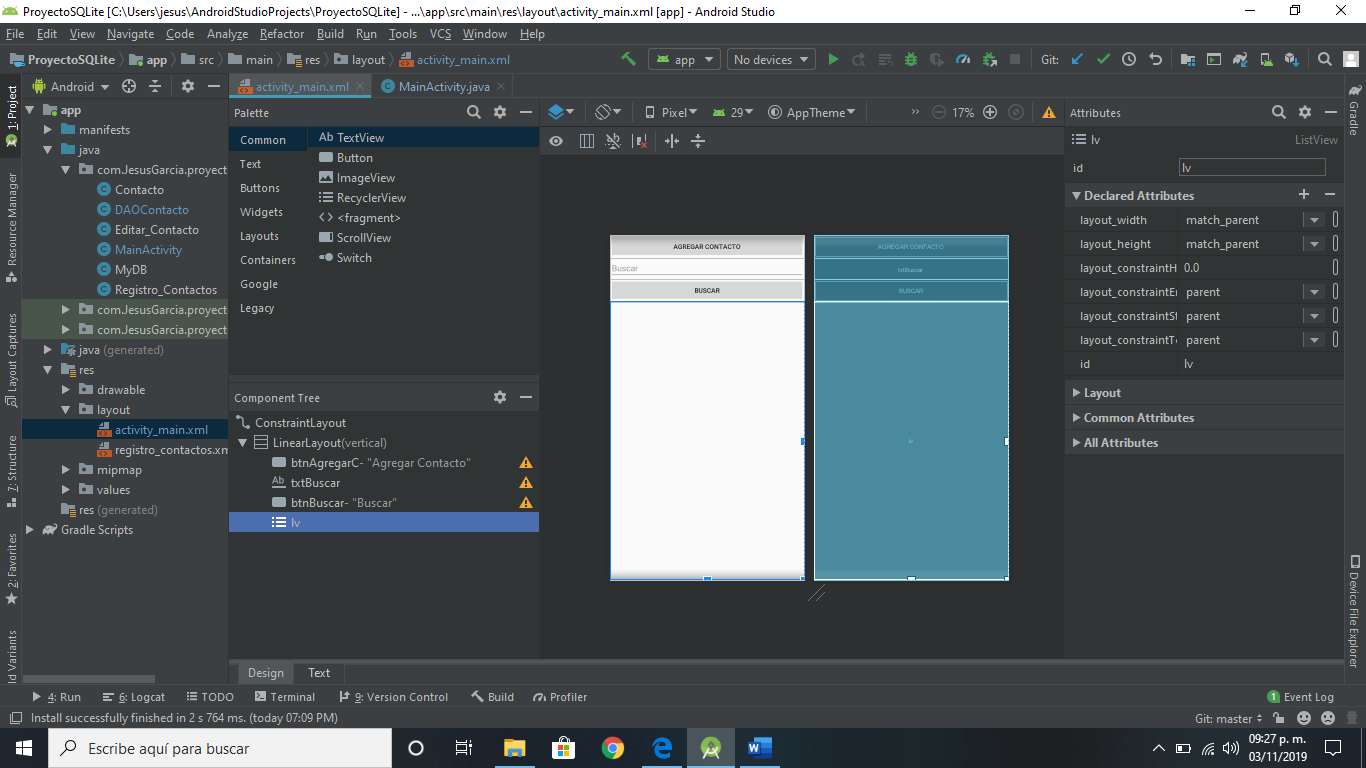
**El método filter 2 cadenas la primera es para el texto a buscar en este caso sería el usuario a buscar y la 2da para saber en que columna se va a buscar el dato anterior.**

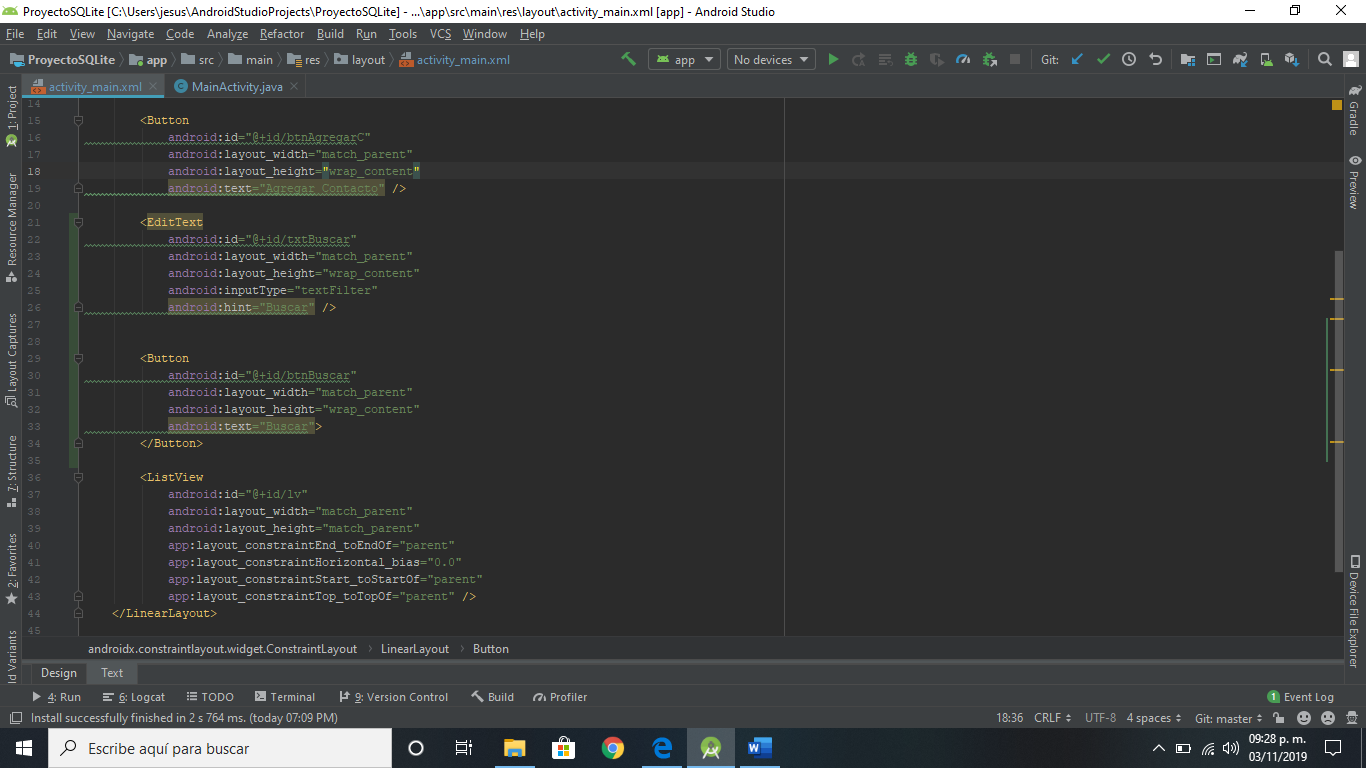
**El método getAll es una lista de tipo contacto en la cual se obtendrá una lista de todos los contactos que se encuentren en la tabla contactos de la base de datos.**



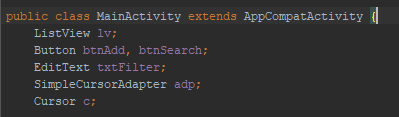


**Creamos la actividad principal (MainActivity) con su respectivo layout (activity\_main). El layout activity\_main se agrego 2 botones el botón de agregar contacto y el botón de buscar, se agrego una caja de texto donde se ingresará los datos a buscar y un listview donde se mostrará una lista de contactos.**

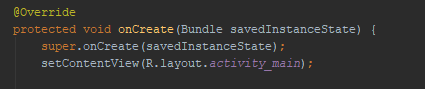




**En el MainActivity creamos las siguientes variables que posteriormente en la misma actividad.**



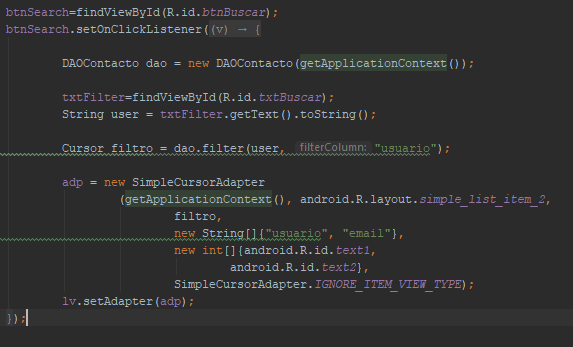
**En el método onCreate hacemos referencia hacia el layout activity\_main para acceder a sus atributos.**



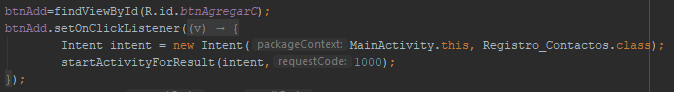
**A continuación, inicializamos la variable btnSearch y le asignamos un escuchador donde cuando lo presionemos se creará una instancia de la clase DAOContacto, después inicializamos la variable txtFilter, luego creamos una variable de tipo string llamada user a la cual le asignaremos el texto que se obtenga de la caja de texto txtBuscar.**

**Después creamos un objeto de tipo cursor llamado filtro al cual se le asignara lo que se obtenga del método filter de la clase DAOContacto.**

**Después al adp (adaptador) que ya teníamos creado le agregamos el objeto de tipo cursor que creamos anteriormente.**



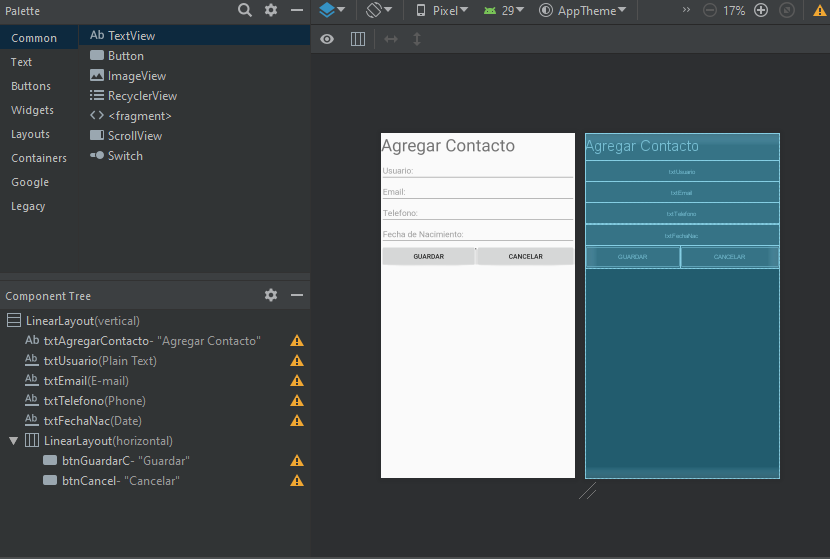
**A continuación, inicializamos el botón btnAdd y le asignamos un escuchador para que cuando se presione el botón se podamos acceder a otra actividad en este caso a Registro\_Contactos**

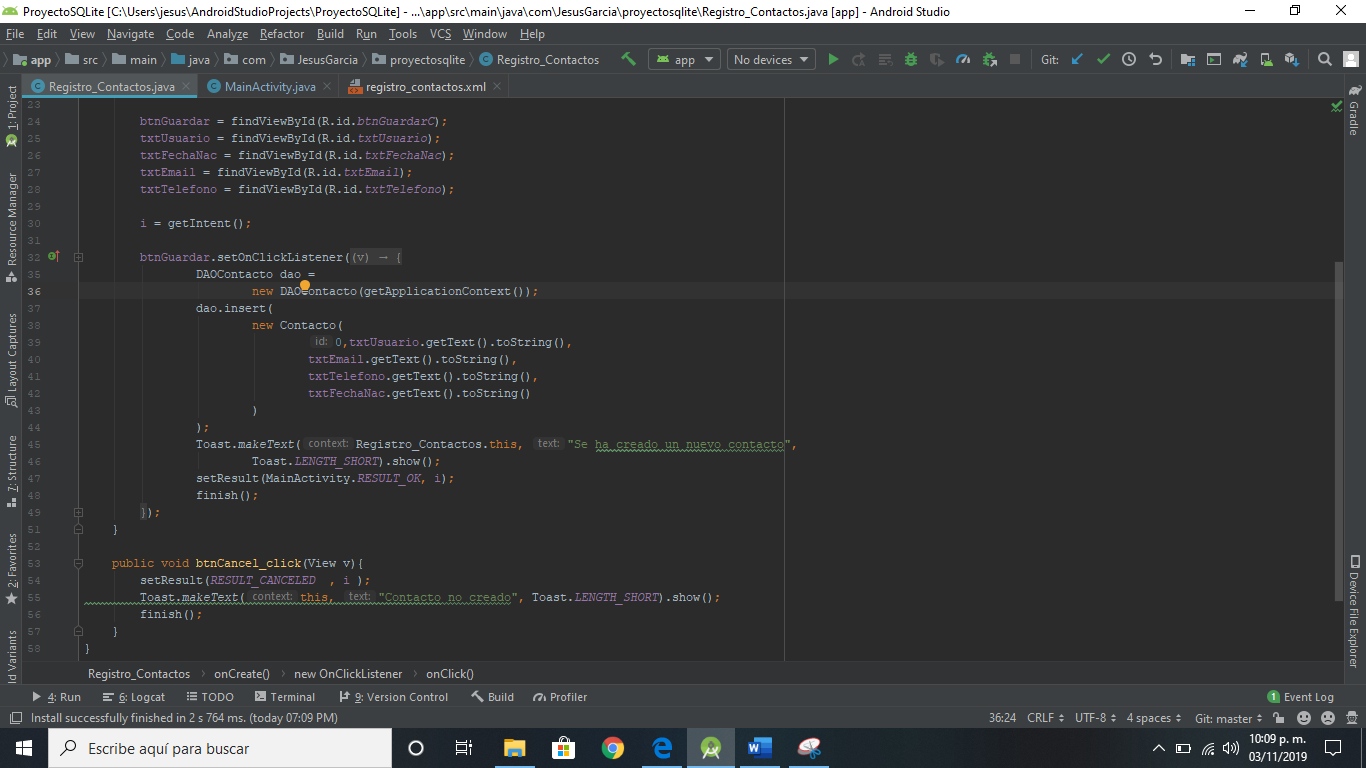


Una vez cuando se haya echo referencia a esta actividad se cargará su respectivo layout que se llama registro\_contactos el cual contiene las cajas de texto para capturar el usuario, email, tel y fechaNac y 2 botones uno para guardar al contacto y otro para cancelar la creación del contacto.

Cuando se presione el botón guardar se hará una instancia de la clase DAOContacto para acceder al método insert e insertar un contacto a la base de datos.

Si se presiona el botón cancelar se retornará a la actividad principal.

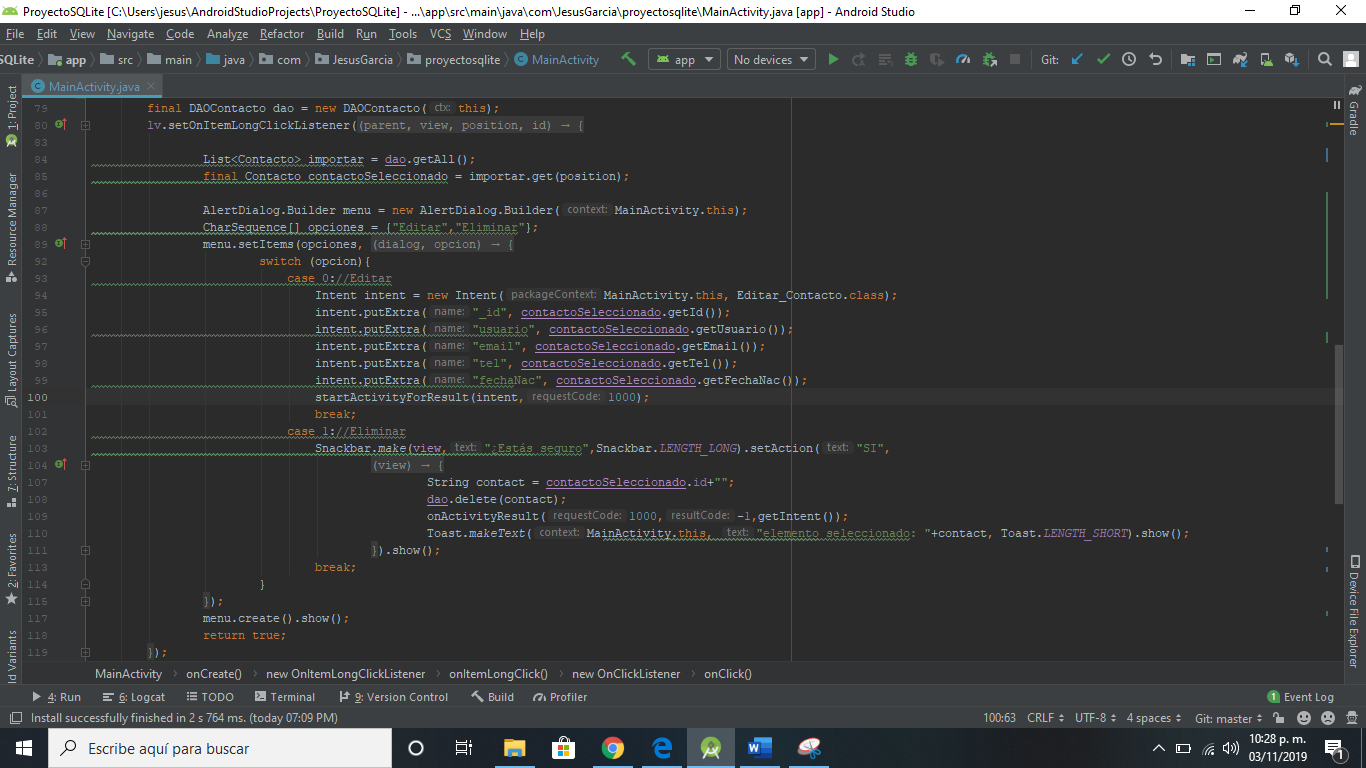




Cuando se haga un click largo en un contacto aparecerá un alertDialog que simulara un menú contextual con las opciones de editar y eliminar a ese contacto seleccionado.

Si se elige la opción de eliminar se pedirá la confirmación de eliminación si se elige si se hará una instancia de la clase DAOContacto para acceder al método delete para eliminar al contacto seleccionado.

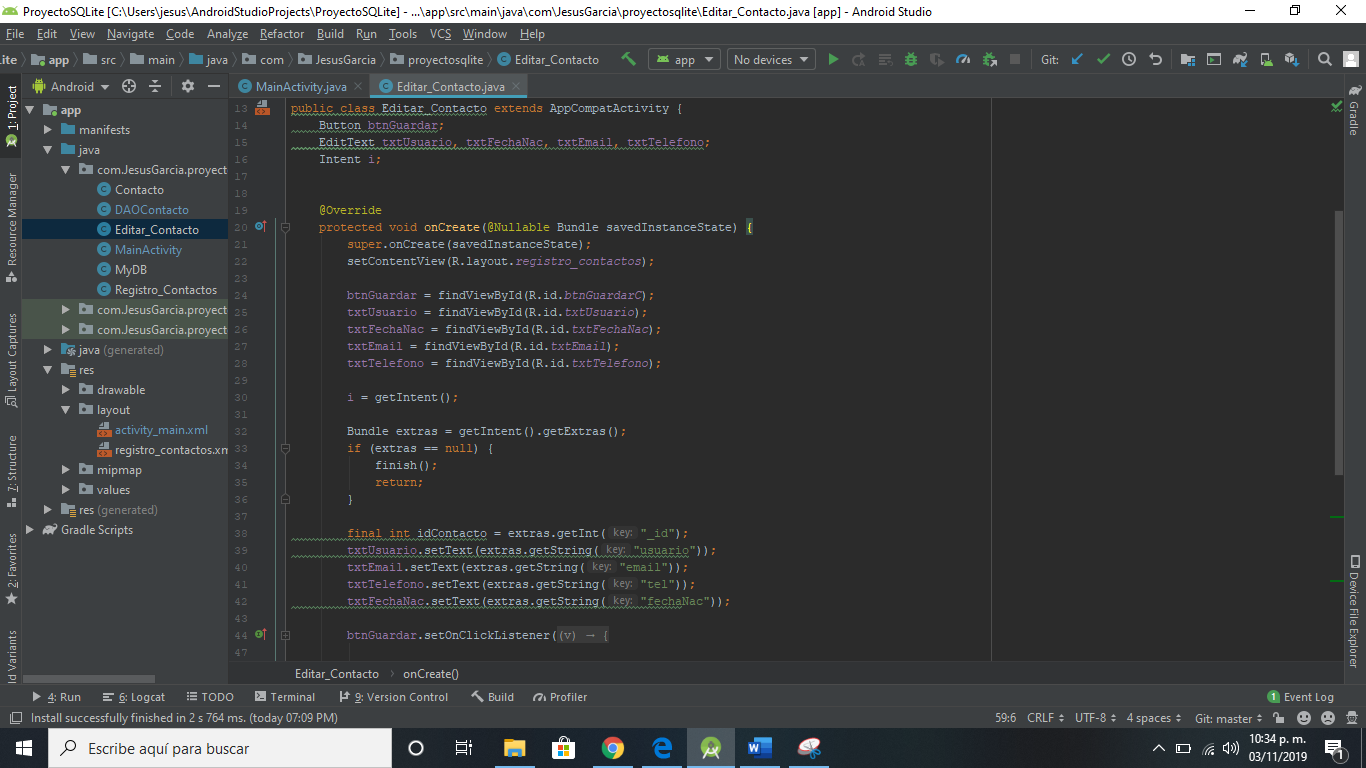
Si se elige la opción editar hacemos una referencia hacia la actividad Editar\_Contacto con el intent pasamos los datos del contacto seleccionado y iniciamos la actividad.

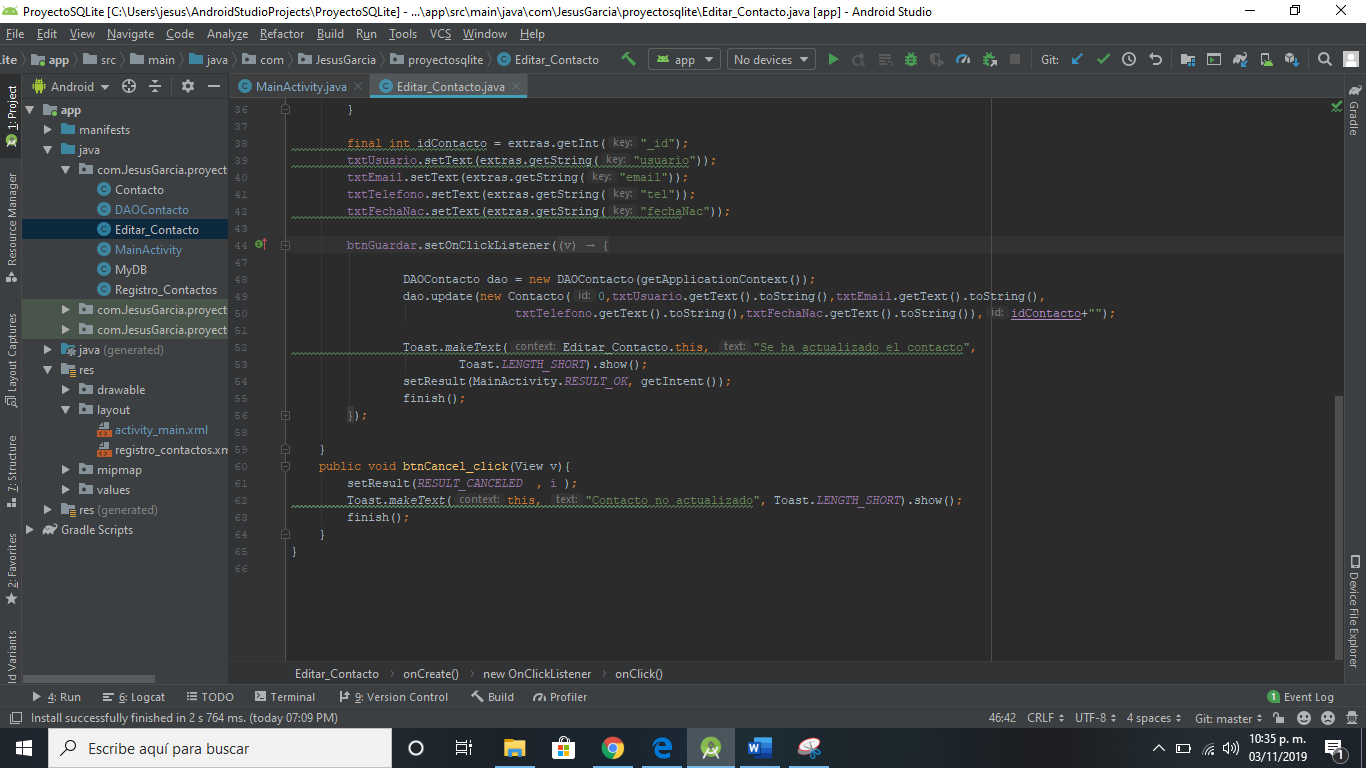


En la actividad Editar\_Contacto recuperamos los datos con Bundle y los asignamos a las cajas de texto del layout registro\_contactos.

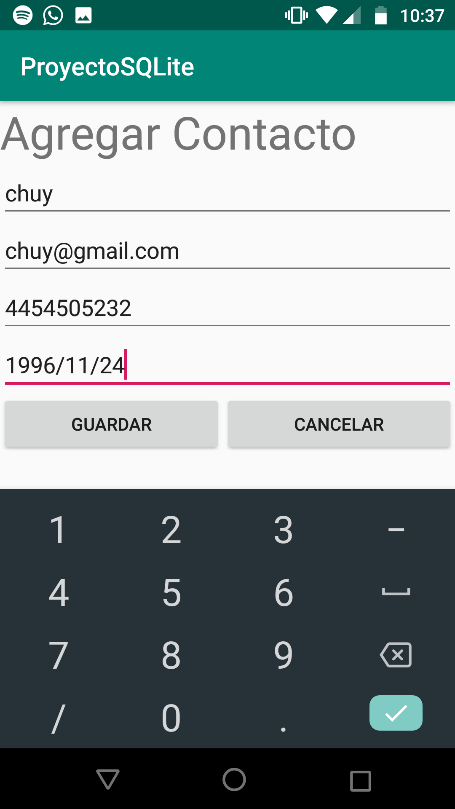
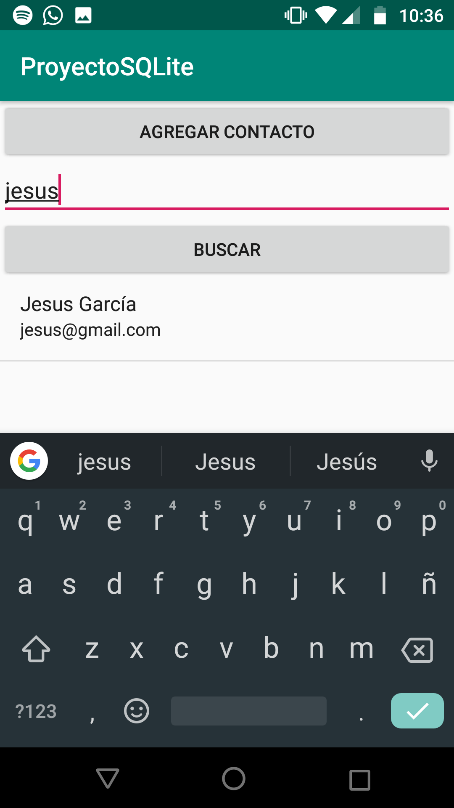
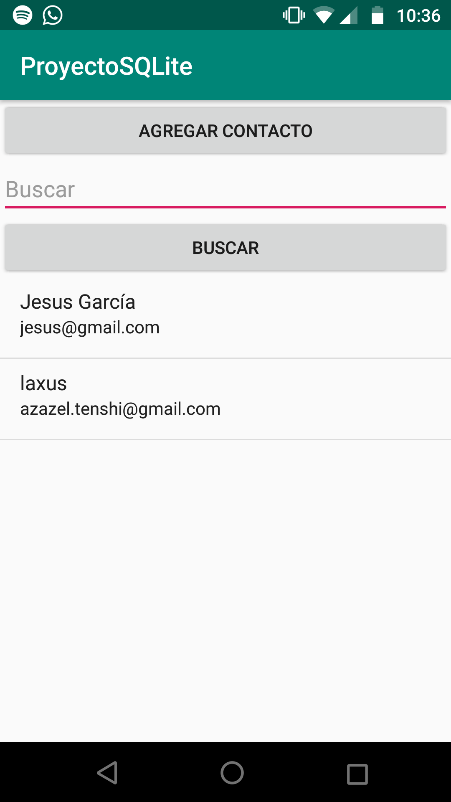
Inicializamos el boton btnGuardar y le asignamos un escuchador para que cuando se presione guarde los datos del contacto.

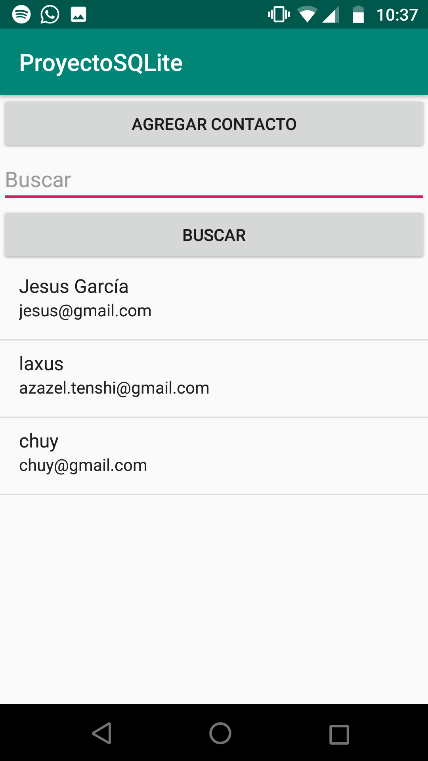
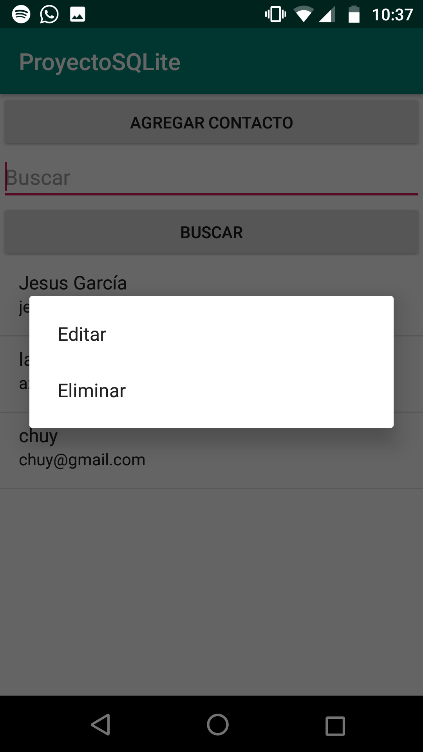
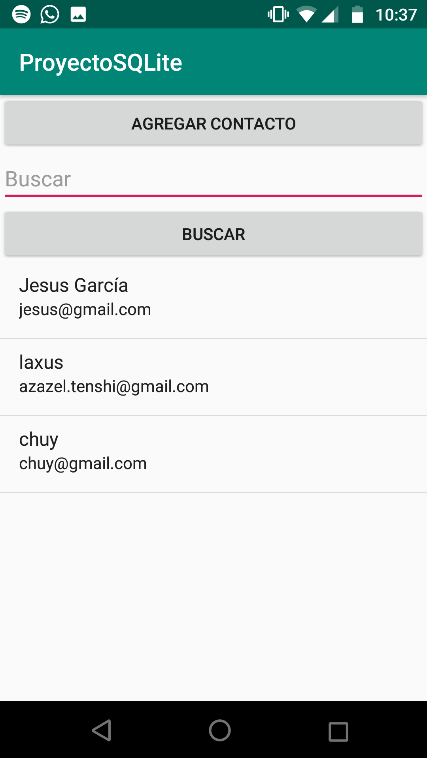
Si se presiona el boton de cancelar se retornara a la actividad principal.

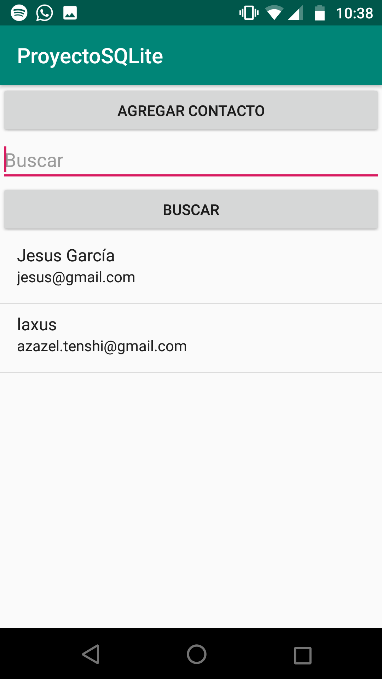
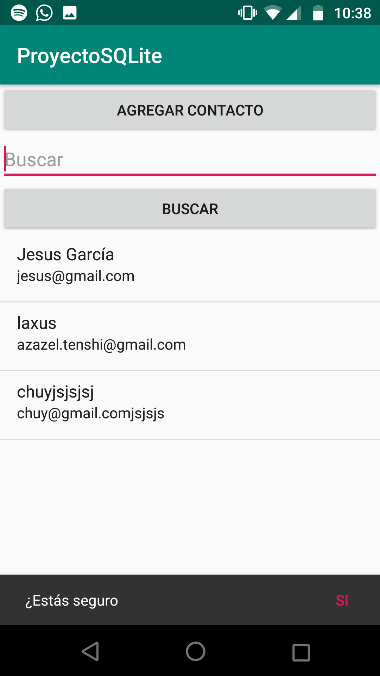
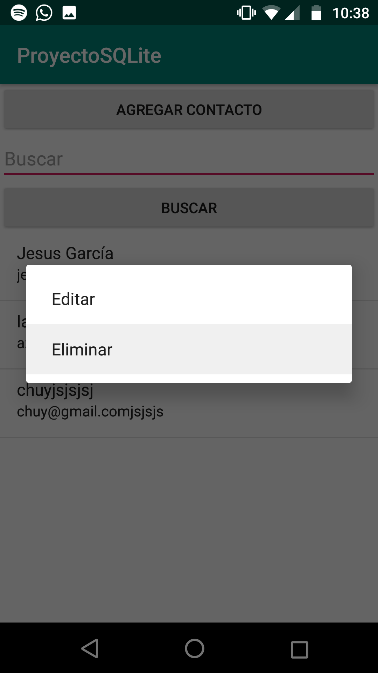




**Resultado**







**Evaluación**

**Conclusiones**

**Bibliografía**