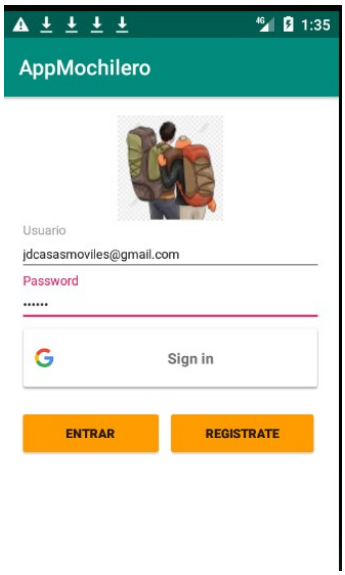
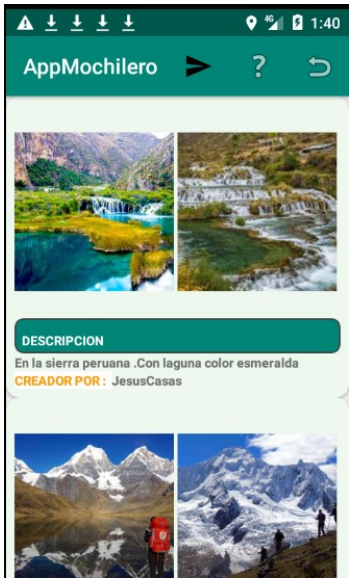
Prueba senior : Tomas J. Casas Rodriguez

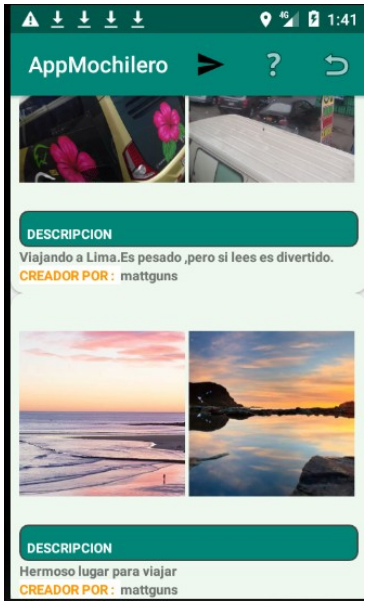
1. **Desarrolle una aplicación para compartir lugares en común, la idea es que puedas registrarte e iniciar sesión en la app, compartir lugares que has conocido con otras personas, la app debe contar con un feed donde se pueden ver los lugares que otras personas han publicado.**

Inicio de sesión

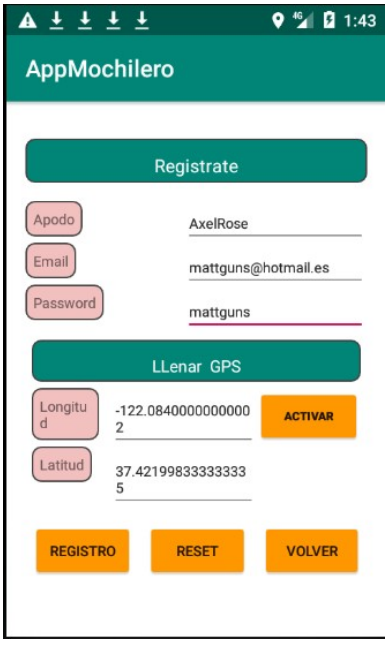


Al entrar se tiene el feed donde se oberva las publicaciones hechas de otros usuarios donde se observa las fotos de los lugares,la descripcion de estos y por quien fue creado .

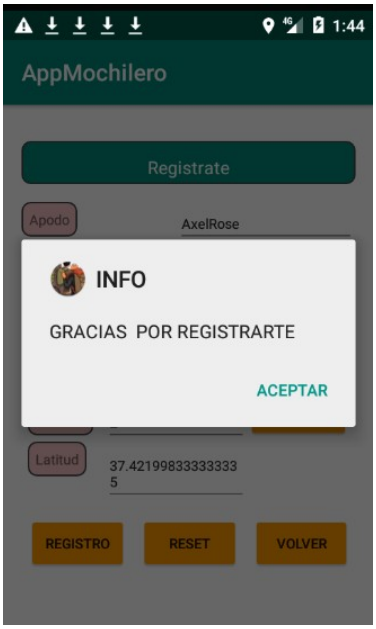




Registro



Nuevo usuario registrado

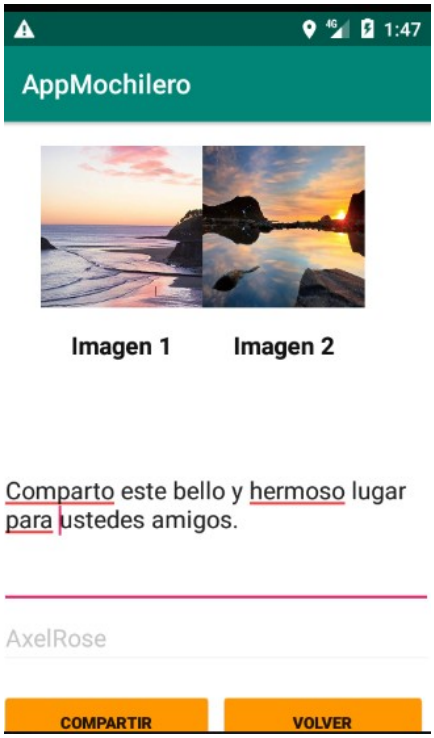


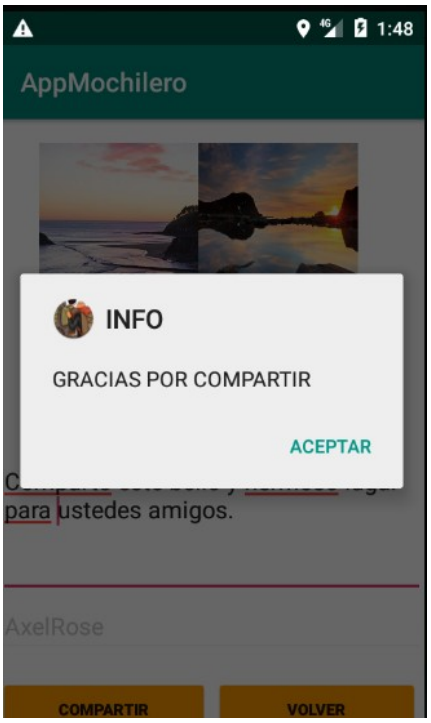
Compartir lugares ,se agrega 2 fotos de tu galeria y la descripcion del lugar.Y se comparte

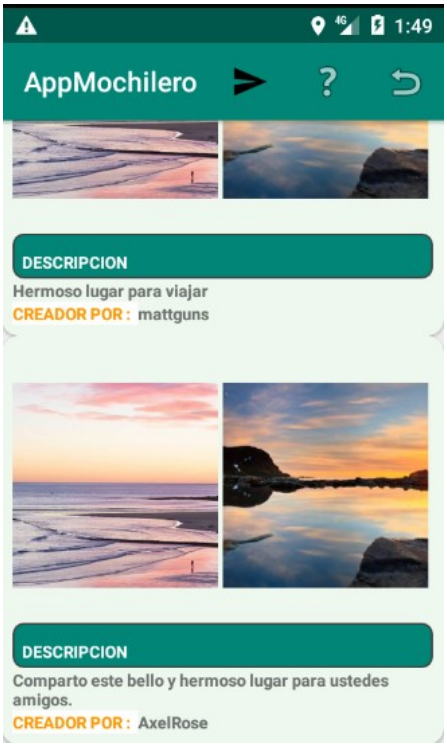
Se agrega fotos de la galeria y se llena la descripcion. Y se comparte

Al dar click en compartir esperar hasta que se muestre el mensaje de gracias por compartir.

Como se observa nuestra publicacion hecha se muestra en el feed.



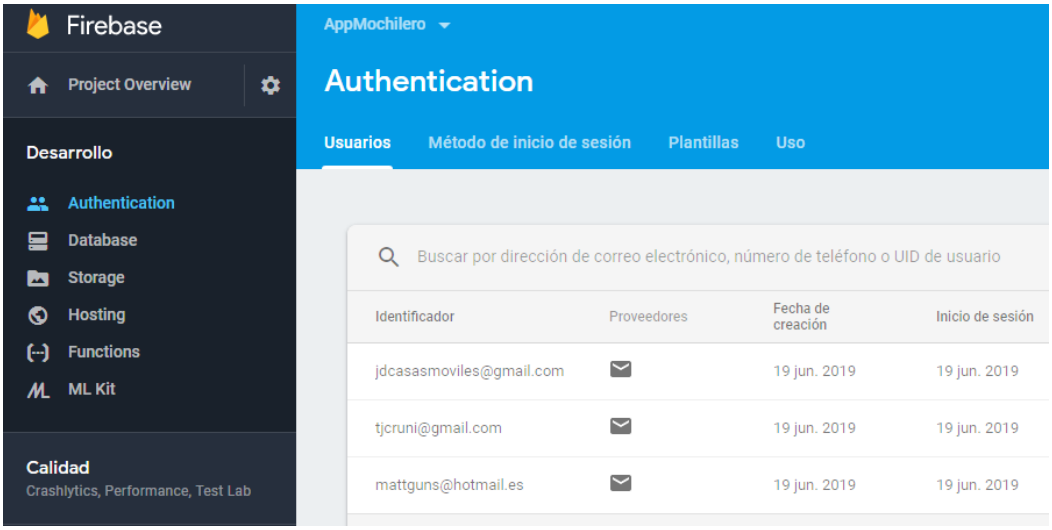


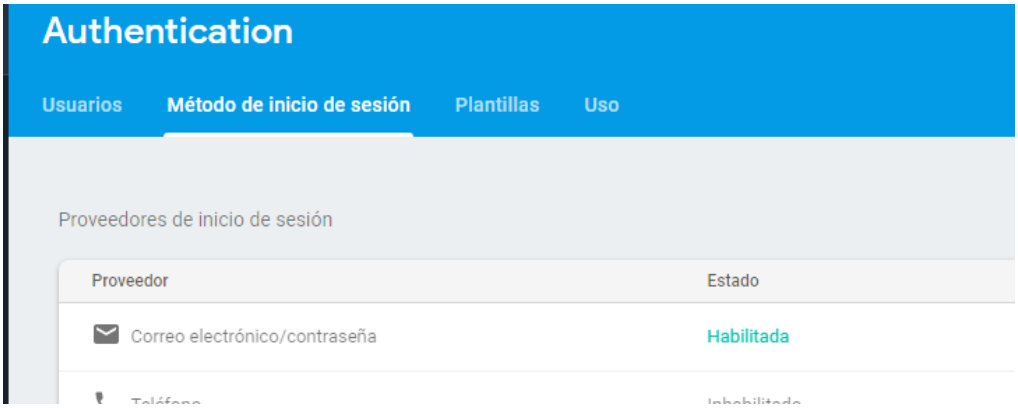


2. **La entidad lugares o Place debe contar con los siguientes campos**
   - **a. Id**
   - **b. Descripción del lugar**
   - **c. Array de fotos (máximo 2)**
   - **d. createdAt**

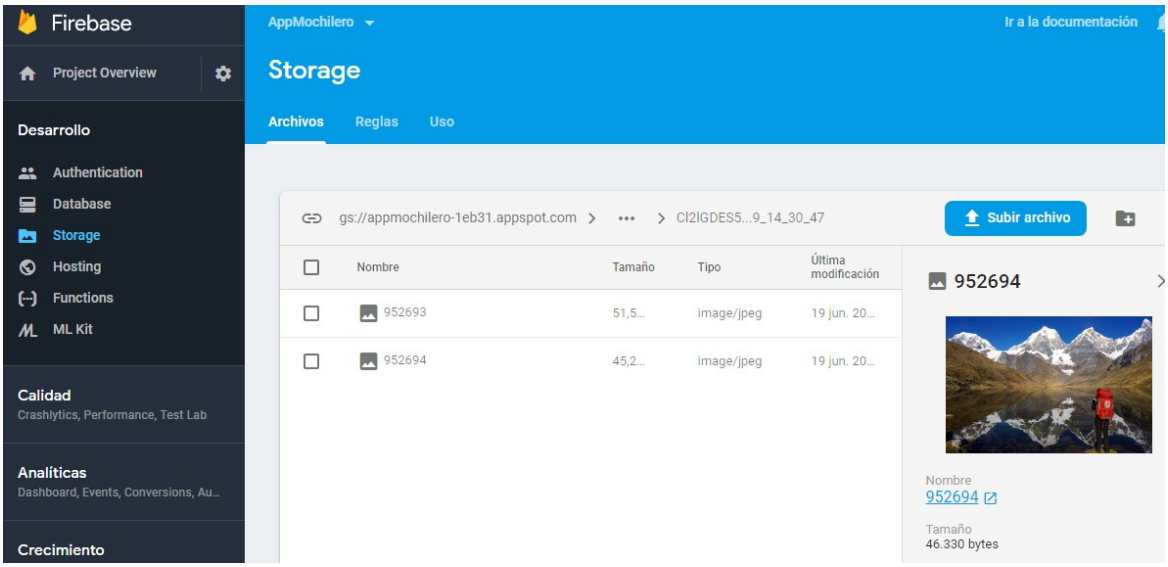
3. Toda la data debe ser almacenada en el producto firestore de firebase

4. Usar producto authentication para gestionar la autenticación/registro :Esta seccion se hace mediante el uso de autenticacion de correo.

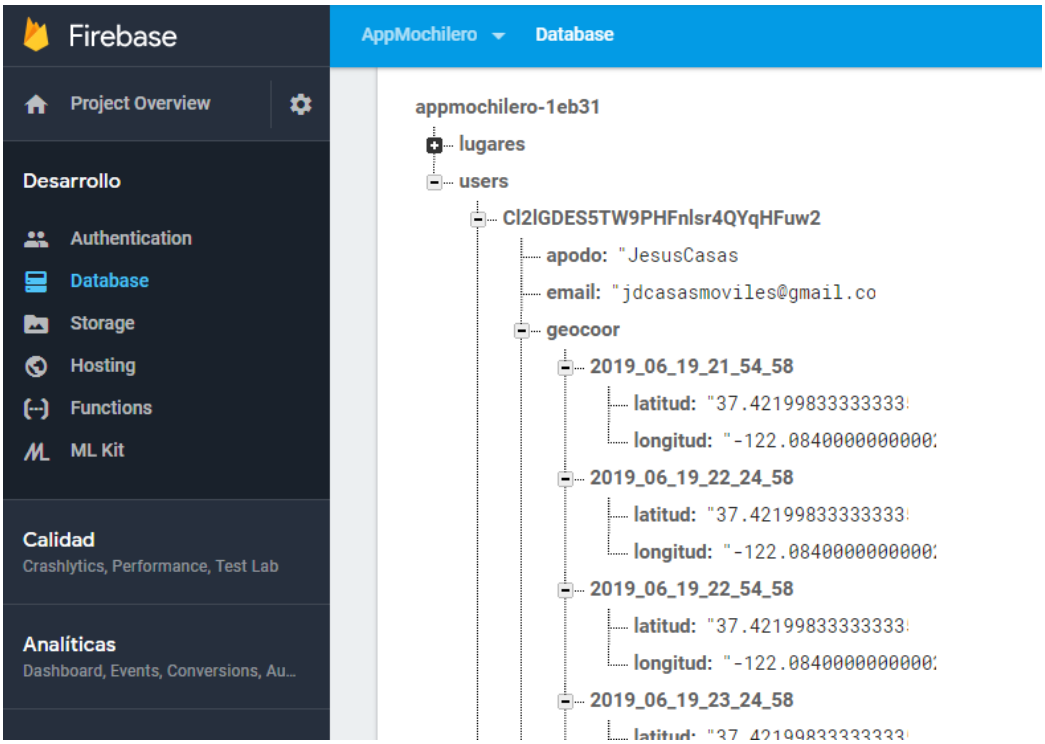




5. Las fotos deben almacenarse con cloud storage :Las imagenes se almacenan en storage.



6. Se debe guardar un registro de la localización del dispositivo, el cual debe enviar la localización cada 30min, debe ser guardada en la entidad del usuario.



Para esta seccion se uso un timer que se ejecuta cada *1800000 milisegundos que equivale a 30 minutos,una vez pasado el tiempo se agrega las nuevas coordenadas de la posicion del dispositivo y se ejecuta de nuevo el timer.*

```
//1 segundos=1 000
//30min=1800segundos=1800000
    private void countDownTimer(){
        new CountDownTimer( millisInFuture: 1800000,  countDownInterval: 1000) {
            public void onTick(long millisUntilFinished) {
            }
            public void onFinish() {
                DatabaseReference mDatabase= FirebaseDatabase.getInstance().getReference();
                DateFormat fecha=new java.text.SimpleDateFormat( pattern: "yyyy_MM_dd_HH_mm_ss");
                mDatabase.child( "users" ).child( id ).child("geocoor").child(fecha.format(new java.util.Date())).child("latitud").setValue(get
                mDatabase.child( "users" ).child( id ).child("geocoor").child(fecha.format(new java.util.Date())).child("longitud").setValue(ge
                countDownTimer();
            }
        }.start();
    }
```

7. Utilizar algún ORM para gestionar la persistencia :Se prefirio usar pojos que estan en la parte de modelos.

8. Puntos a ser evaluados a. Uso de algún patrón de diseño :Se uso el patron MVC ,este patron tiene tres capas las vistas que se refiere a la interfaz grafica,controlador se refiere a las funcionalidades principales que hace la app y modelo donde esta el esquema se usa para las base de datos no relacionales que se usa tambien en esta parte se comportan como un pojo.

b. Código limpio y ordenado : Es lo mas simple posible y tiene anotaciones donde se requiera mejor compresion.

c. Punto adicional pruebas unitarias :Cuando entran en alguna excepcion la app te informa lo que ocurre o porque no se dio tal operación.

9. El código debe subirse a github y compartir el link del proyecto y enviar el apk para su prueba.

Link GitHub : https://github.com/jdcasasmoviles/AppMochilero

10. Entregables

a. Apk : El apk se encuentra en este link :
https://drive.google.com/file/d/1DGoFuPHr8_HX1rsG4N_I-0Im_n-9WQCQ/view?usp=sharing

b. Link del proyecto en github : https://github.com/jdcasasmoviles/AppMochilero

c. Dentro del proyecto github explicando cada una de las capas que utilizo para desarrollar el proyecto :En esta parte se sube este pdf donde se explica de forma resumida las clases de la app.

CAPA DE VISTA :

MainActivity : En esta clase se hace el cargado de datos de la entidad lugares y se los pasa a la instancia RecyclerAdapterLugares adapter_ra.

DialogKachuelo : Esta clase sirve para mostrar los dialogos informativos que aparecen en la app.

RecyclerAdapterLugares :Se llena los card view con la informacion obtenida.

CAPA DE CONTROLADOR :

CompartirActivity : Se hace la subida de archivos mediante el uso de un *metodo recursivo, finaliza cuendo todas las imagenes se hallan subido al s rage y se agrega los datos a la entidad lugares.*

RegistroActivity : Se agrega usuarios a la entidad users, con el uso unico de un solo correo,el alias o apodo,password y las coordenadas.

Geolocalizacion : Esta clase se usa para llenar de forma automatica el formulario de registro y hacer activar el GPS.Ademas tiene la funcionalidad del timer que cada 30 min agrega la nueva posicion del dispositivo en la entidad usuarios de la database.

LoginActivity:Esta clase hace la autenticacion de usuarios mediante FirebaseAuth

CAPA DE MODELO :

Lugares: Esta clase funciona como un pojo y ademas expone el modelo de la entidad lugares,como sus campos mas resaltantes: D escripcion, imagen1, imagen2 y creadoPor.

Usuarios: Esta clase funciona como un pojo y ademas expone el modelo de la entidad users,como sus campos mas resaltantes: Apodo, email, password, latitud y longitud.

.

CardView