



UNIVERSIDAD POLITÉCNICA DE PACHUCA

DOCUMENTACIÓN

APLICACION MOVIL

PRESENTA:

VELÁZQUEZ PÉREZ SERGIO JESÚS
GRANADA TEODORO JOSÉ ANTONIO
HERNANDEZ RAMIREZ DORA ANELLY

INGENIERÍA DE SOFTWARE

DIRIJIDO POR: ING. KEVIN EMMANUEL JARAMILLO LIEVANO

Índice general

1. Introducción	4
1.1. Planteamiento del problema	4
1.2. Objetivo general	4
1.3. Objetivos específicos	4
2. Justificación	5
2.1. ¿Por qué una Aplicación móvil?	5
2.2. ¿Dónde se aplicará?	5
3. Marco teórico	6
3.1. Herramientas utilizadas	6
3.1.1. Java	6
3.1.2. android	6
3.1.3. Sql lite	6
3.1.4. material design	6
3.1.5. GIT	6
3.1.6. UML	6
3.1.7. Mock up	7
4. metodologia	8
4.1. Analisis	8
4.1.1. Requisitos funcionales	8
4.1.2. Requisitos no funcionales	8
4.2. diseño	9
4.2.1. Diagrama casos de uso	9
4.2.2. mockup	10
4.2.3. diagrama de actividades	11
4.2.4. diagrama de secuencias	12
4.3. programación	12
4.3.1. Código LoginActivity	14
5. Resultados y conclusiones	15

Índice de figuras

4.1. diagrama casos de uso	9
4.2. mockoup	10
4.3. mockoup	10
4.4. mockoup	11
4.5. diagrama de actividades	11
4.6. diagrama de secuencias	12
4.7. estructura del proyecto	13
4.8. layouts	14
4.9. Código	14

Capítulo 1

Introducción

1.1. Planteamiento del problema

En un restaurante surge la necesidad de hacer reservaciones de una manera más tecnológica y rápida ya que últimamente ha tenido un problema con la administración de mesas reservadas.

Otra necesidad del restaurante es que no ha tenido el impacto que desearía tener por lo que necesita dar a conocer sus platillos y promociones para así atraer a más personas al restaurante.

Se propone realizar una aplicación móvil donde el usuario pueda hacer una reservación de una mesa para así poder agilizar el proceso de reservar y que sea de una manera más fácil y rápida para el usuario.

1.2. Objetivo general

Desarrollar una aplicación móvil que permita a los clientes reservar mesas del restaurante y realizar pedidos de comida a domicilio para contribuir al crecimiento del negocio y brindar a sus clientes una forma de adquirir sus platillos sin la necesidad de trasladarse al restaurante.

1.3. Objetivos específicos

- Crear una web service para el desarrollo de la aplicación.
- Conectar la aplicación a una base de datos para el manejo de información
- Desarrollar una interfaz donde el usuario pueda ver promociones sin necesidad de registrarse para que el usuario tenga una visión de lo que es el restaurante.

Capítulo 2

Justificación

2.1. ¿Por qué una Aplicación móvil?

El restaurante sabe que mucha gente trabaja o tiene muy poco tiempo para ir a el restaurante para comer, lo más importante para el restaurante son sus clientes, por esa razón el restaurante necesita que el cliente pueda pedir comida a domicilio o a su trabajo. Una forma rápida y muy eficiente en que los clientes del restaurante puedan reservar una mesa o pedir comida a domicilio es a partir de una aplicación móvil en Android ya que la mayoría de los clientes del restaurante cuenta con un dispositivo móvil.

2.2. ¿Dónde se aplicará?

La aplicación se puede implementar en restaurantes que cuenten con la opción de apartar mesas a sus clientes y que ofrezcan servicio a domicilio.

Capítulo 3

Marco teórico

3.1. Herramientas utilizadas

3.1.1. Java

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. [7, Oracle Corporation. (2018)]

3.1.2. android

Android ofrece un completo framework de aplicaciones que te permite crear apps y juegos innovadores para dispositivos móviles en un entorno de lenguaje Java. Los documentos que se indican en la barra de navegación izquierda proporcionan detalles acerca de cómo crear apps usando diferentes API de Android. [6, Android. (2018)]

3.1.3. Sql lite

SQLite es un motor de base de datos SQL incorporado. A diferencia de la mayoría de las otras bases de datos SQL, SQLite no tiene un proceso de servidor por separado. SQLite lee y escribe directamente en archivos de disco ordinarios. Una base de datos SQL completa con múltiples tablas, índices, disparadores y vistas, está contenida en un solo archivo de disco. [1, sqlite. (2017)]

3.1.4. material design

Material Design es un sistema unificado que combina teoría, recursos y herramientas para crear experiencias digitales. [2, Material Design. (2017)]

3.1.5. GIT

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. [3, git hub,(2018)]

3.1.6. UML

El lenguaje UML es un estandar de OMG diseñado para visualizar, especificar, construir y documentar software orientado a objetos. [5, Grady Booch. (2015)]]

3.1.7. Mock up

Los mockups son muy útiles para que, cuando empieces el diseño final, no pierdas de vista el objetivo. Los bocetos sirven como brújulas durante el proceso de diseño. En Canva, tienes muchas herramientas de diseño para crear mockups de sitios web, de aplicaciones o de lo que quieras. No importa si estás trabajando como freelancer o si necesitas generar una gran cantidad de piezas, en Canva es muy rápido y fácil crear excelentes mockups para que tengas más tiempo para el diseño final. [4, canva. (2018)]

Capítulo 4

metodologia

4.1. Analisis

4.1.1. Requisitos funcionales

- 1.-El sistema permitirá visualizar a todos los usuarios los precios y promociones de los platillos del restaurante in la necesidad de loguearse en la aplicación.
- 2.-El sistema permitirá solo a los usuarios registrados la autorización de poder reservar una mesa hasta un día antes.
- 3.-el sistema permitirá solo a los usuarios registrados la autorización de poder hacer pedidos a domicilio.

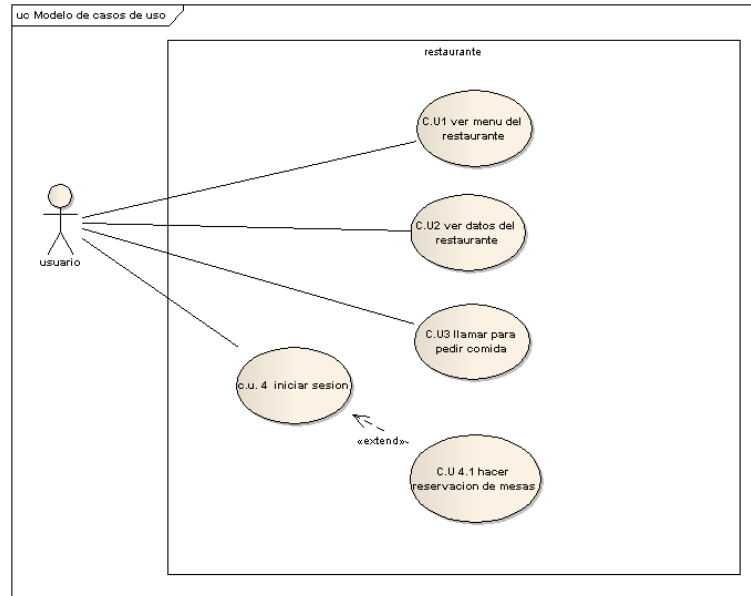
4.1.2. Requisitos no funcionales

- 1.-La aplicación debe ser capaz de proporcionar mensajes de error a los usuarios
- 2.- El sistema no revelara a sus operadores otros datos personales de los clientes..
- 3.-La aplicación debe tener una interfaz basada en material design

4.2. diseño

4.2.1. Diagrama casos de uso

Figura 4.1: diagrama casos de uso



4.2.2. mockup

Figura 4.2: mockup

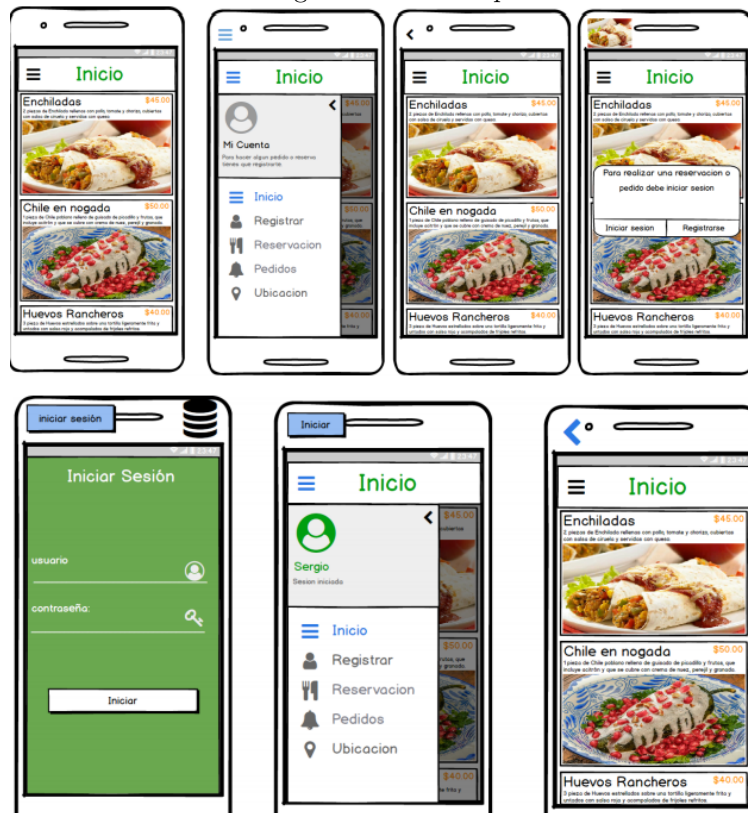


Figura 4.3: mockup

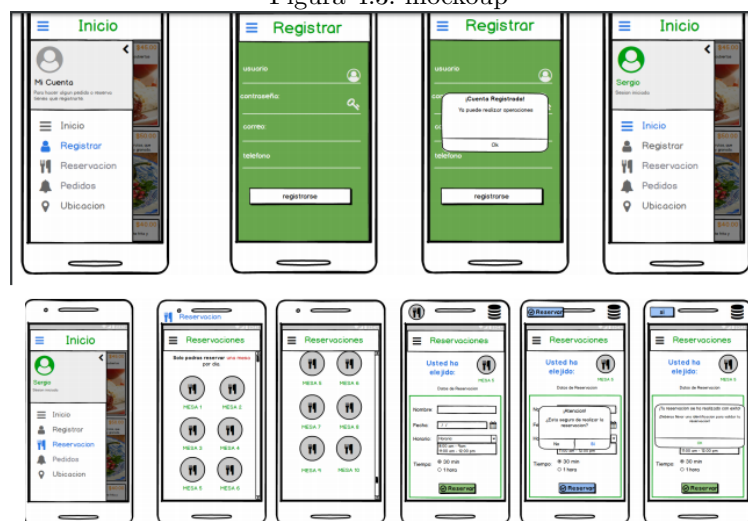
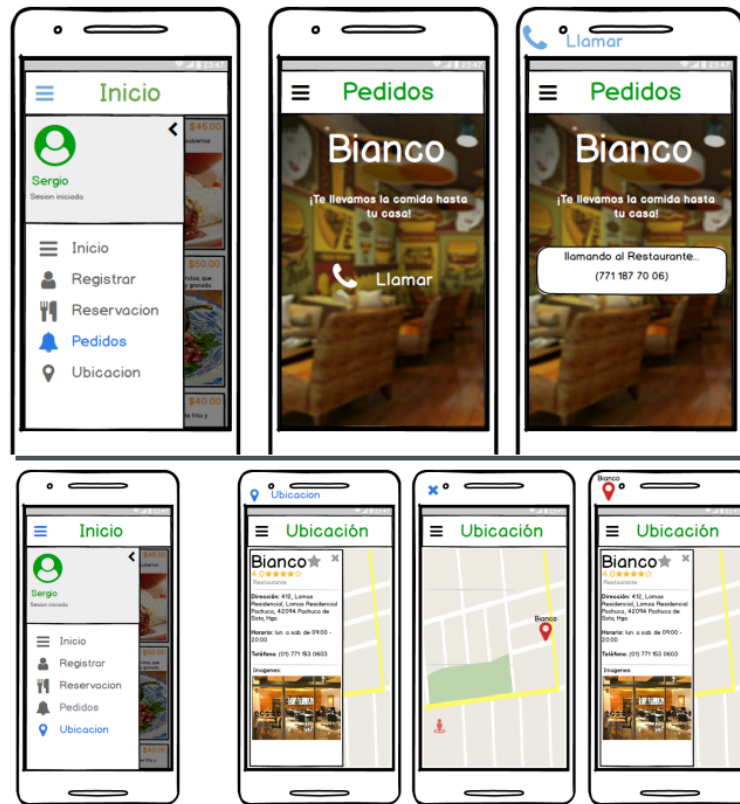
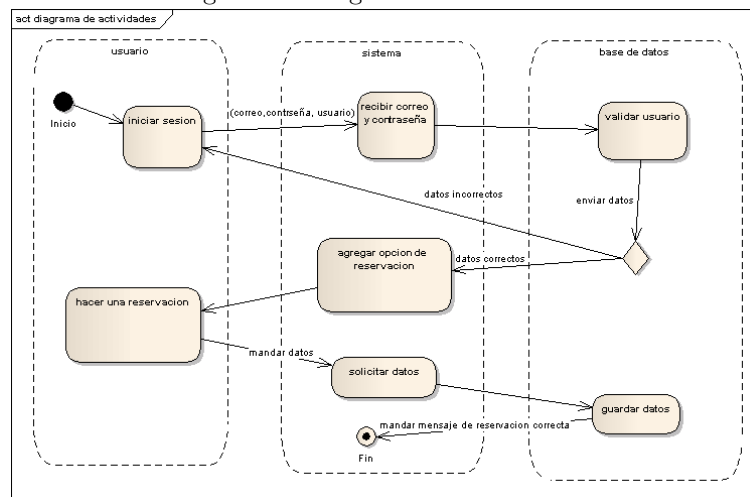


Figura 4.4: mockoup



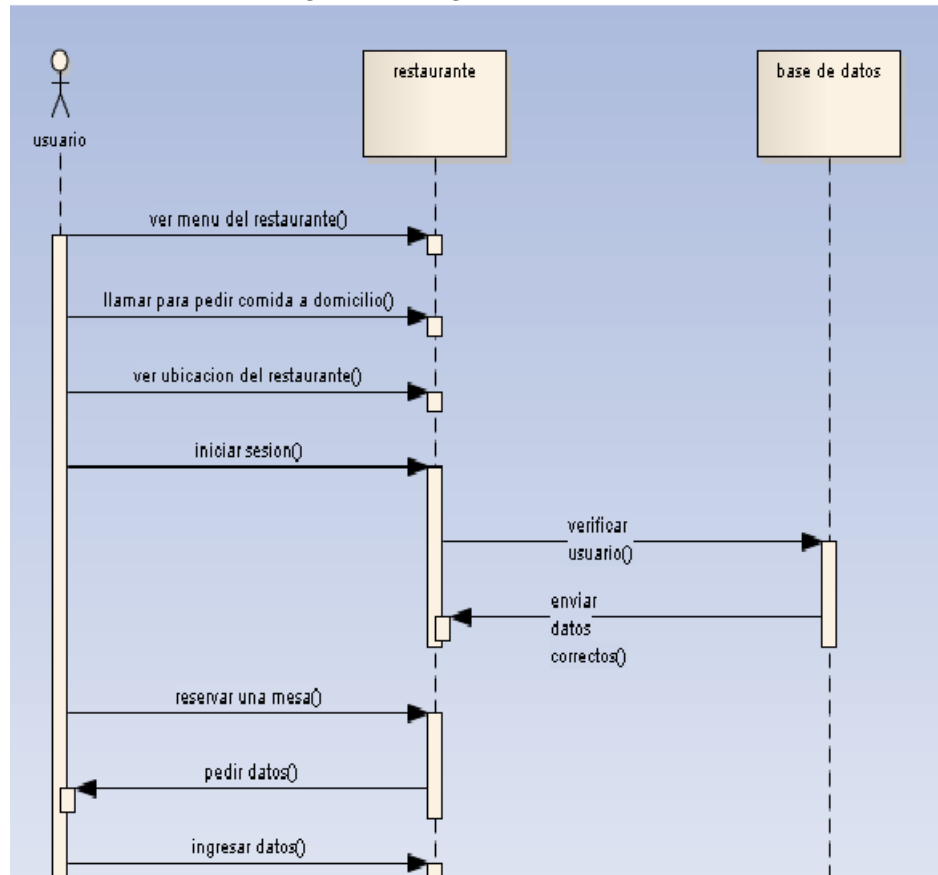
4.2.3. diagrama de actividades

Figura 4.5: diagrama de actividades



4.2.4. diagrama de secuencias

Figura 4.6: diagrama de secuencias



4.3. programación

Para realizar la programación de esta aplicación utilizaremos Android Studio para el diseño y desarrollo de la aplicación móvil, y firebase para la gestión de la base de datos.

Figura 4.7: estructura del proyecto

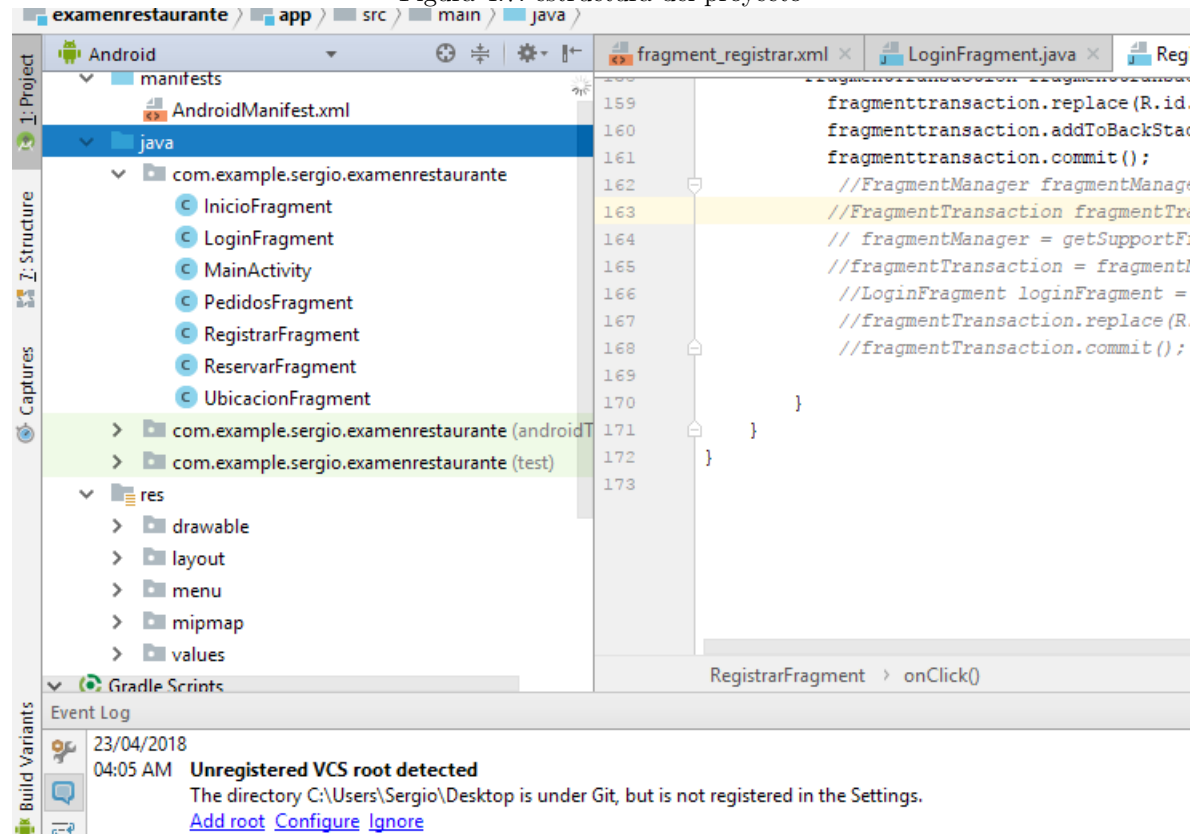
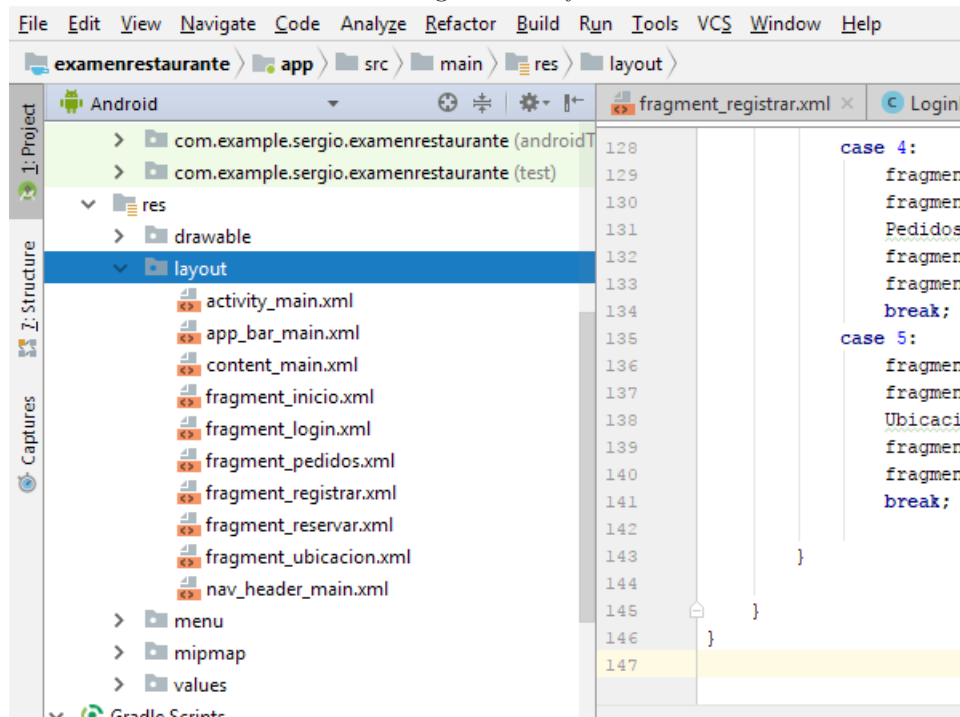


Figura 4.8: layouts



4.3.1. Codigo LoginFragment

Esta parte de código conecta nuestra interfaz con la programación que se requiere para el desarrollo del login de nuestra aplicación

Figura 4.9: Codigo

```
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    View view = inflater.inflate(R.layout.fragment_login, container, attachToRoot false);

    firebaseAuth = FirebaseAuth.getInstance();

    if(firebaseAuth.getCurrentUser() != null){
        Intent i = new Intent( packageContext LoginActivity.this, InicioFragment.class );
        startActivity(new Intent( packageContext LoginActivity.this, InicioFragment.class ));
    }
    editTextcorreo = (EditText) view.findViewById(R.id.editTextcorreo);
    editTextcontrasena = (EditText) view.findViewById(R.id.editTextcontrasena);
    btnlogin = (Button) view.findViewById(R.id.btnlogin);

    progressDialog = new ProgressDialog( context LoginActivity.this);
    btnlogin.setOnClickListener((v) -> {
        Intent i = new Intent( packageContext LoginActivity.this, InicioFragment.class);
        startActivity(i);
    });
    return view;
}

private void userLogin() {
    String correo = editTextcorreo.getText().toString().trim();
    String contrasena = editTextcontrasena.getText().toString().trim();

    if (TextUtils.isEmpty(correo)) {
        Toast.makeText( context LoginActivity.this, text "por favor escribe un correo", Toast.LENGTH_SHORT).show();
        // Toast.makeText(this, "hij", Toast.LENGTH_SHORT).show();
        return;
    }
}
```

4.3.2. Codigo RegistrarFragment

Esta parte de código registra los nuevos usuarios que tendrá nuestra aplicación y los registra en un servidor web.

Figura 4.10: Codigo

```
private void registrarusuario(){
    String correo = editTextCorreo.getText().toString().trim();
    String contrasena = editTextcontrasena.getText().toString().trim();

    if(TextUtils.isEmpty(correo)){
        Toast.makeText(getContext(), "por favor escribe un correo", Toast.LENGTH_SHORT).show();
        // Toast.makeText(this, "jhj", Toast.LENGTH_SHORT).show();
        return;
    }

    if(TextUtils.isEmpty(contrasena)){
        Toast.makeText(getContext(), "por favor escribe una contraseña", Toast.LENGTH_SHORT).show();
        // Toast.makeText(this, "jhj", Toast.LENGTH_SHORT).show();
        return;
    }

    progressDialog.setMessage("registrar usuario");
    progressDialog.show();

    firebaseAuth.createUserWithEmailAndPassword(correo, contrasena)
        .addOnCompleteListener((Activity) getContext(), (task) -> {
            if(task.isSuccessful()){
                Toast.makeText(getContext(), "usuario registrado correctamente", Toast.LENGTH_SHORT).show();
            }else{
                Toast.makeText(getContext(), "usuario no registrado intente nuevamente por favor", Toast.LENGTH_SHORT).show();
            }
        });
}
```

Capítulo 5

Resultados y conclusiones

Podemos concluir que el desarrollo de aplicaciones con Android studio es una herramienta muy completa para el desarrollo de aplicaciones nativas, ya que contiene diferentes componentes y herramientas que nos ayudan a un mejor diseño y funcionalidad de la aplicación. Sin embargo tiene un nivel complicado para la diseño y desarrollo de aplicaciones ya que puede llegar a ser complicada la implementación de lo que si quiera desarrollar.

Para el desarrollo de esta aplicación hicimos uso de los diferentes componentes que proporciona Android Studio para el diseño y programación de la aplicación como el uso de fragments, activities, y elementos visuales, conectándolo con una base de datos desde firebase que lleva el control de los datos que dicha aplicación utiliza.

Bibliografía

- [1] sqllite. (2017). Acrca de SqlLite. abril 20, 2018, de SQLite Sitio web: <https://www.sqlite.org/about.html>
- [2] Material Design. (2017). Material Design. abril 20, 2018, de Material Design Sitio web: <https://material.io/>
- [3] git hub. (2018). ¿Que es git hub?. abril 20,2018, de git hub Sitio web: <https://github.com/LuchoCastillo/Conociendo-GitHub/blob/master/tutorial/data/introduccion.rst>
- [4] canva. (2018). Crea tus mock up y dale forma a tus ideas. canva, de abril 20,2018 Sitio web: https://www.canva.com/es_mx/crear/mockups/
- [5] Grady Booch. (2015). UML. abril 21 2018, de UML Sitio web: <http://elvex.ugr.es/decsai/java/pdf/3E-UML.pdf>
- [6] Android. (2018). Introduccion a android. Abril 21, 2018, de Android Sitio web: <https://developer.android.com/guide/index.html?hl=es-419>
- [7] Oracle Corporation. (2018). ¿que es Java?. abril 21, 2018, de Oracle Corporation Sitio web: https://www.java.com/es/download/faq/whatis_java.xml