Proyecto Final de Grado Desarrollo de Aplicaciones Multiplataforma

FitWork: Mantente en forma

Álvaro Tejero Nuño 2º DAM

Tutor: Anahí Mula de la Banda

Desarrollo de Aplicaciones Multiplataforma

Cada vez hay más personas que quieren dar un cambio de salud y físico en su vida, pero no saben cómo empezar, ¿qué debo hacer para empezar?, ¿entrenar en casa o ir a un gimnasio?, ¿es obligatorio hacer dieta? Y una larga lista de etc. Todo esto y más son muchas de las dudas que las personas se preguntan a la hora de ponerse a entrenar.

El objetivo de esta aplicación es ayudar a principiantes para que tengan una idea de cómo empezar a cuidarse, las medidas más adecuadas para cada uno y una serie de tablas personalizadas para los objetivos de cada persona, ya sea desde perder peso a ganar más resistencia. Además, la aplicación contará con un foro donde muchos de nuestros profesionales podrán contestar a las preguntas de los usuarios y donde los usuarios puedan ayudarse y resolver dudas entre sí.

Sobre todo, el primer paso para ponerse manos a la obra es tener ganas, no desmotivarse y no rendirse, los cambios no surgen de un día para otro.

Por supuesto esta aplicación no solo es para principiantes, la aplicación es válida para cualquier persona ya que hay tablas de ejercicios de distintos niveles de dificultad, así como de duración de tiempo.

More and more people want to make a health and physical change in their lives, but they don't know how to start, what should I do to get started? Do I train at home or go to a gym? Is the diet obligatory? And a long list of etc. All this and more are many of the questions that people ask when they start training.

The objetive of this application is to help beginners to start taking care themselves, the most appropriate steps for each and a series of tables customized to each person's goals, from losing weight to gaining more strength. In addition, the application will have a forum where many of our professionals will be able to answer user's questions and where users can help each other and resolve any doubts.

Above all, the firts step to in getting down to work is to have the will, dont lose your motivation and dont give up, the changes dont come overnight.

Of course this application isnt only for beginners, the application is valid for anyone, there are tables of exercises of different levels of difficulty, as well as duration of time.

Tabla de **contenido**

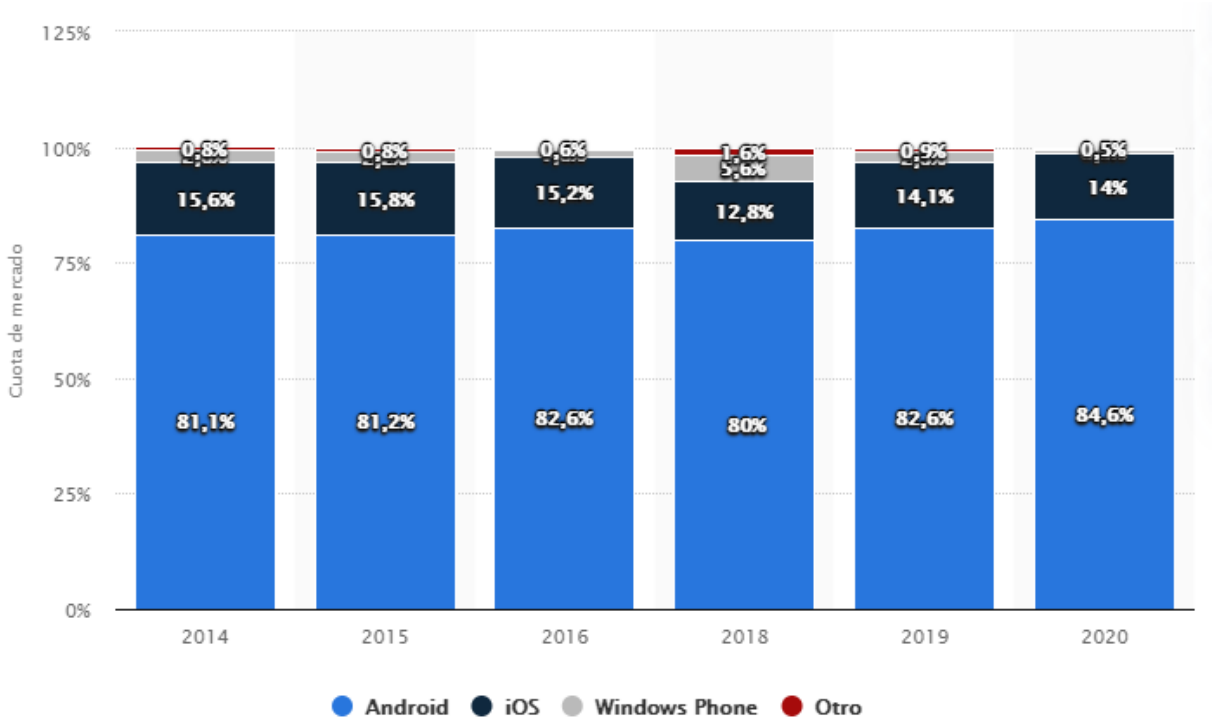
1. INTRODUCCIÓN

1.1 Justificación del Proyecto

El proyecto consiste en una aplicación Android desarrollada en Java destinada al uso del ámbito deportivo, en ella se encontrarán perfiles de tablas personalizadas para cada tipo de usuario y objetivo.

La aplicación tendrá un apartado de foro donde los usuarios pueden comunicarse entre sí para preguntarse dudas sobre ejercicios, dietas y todo lo relacionado con lo deportivo, además muchos de nuestros profesionales también estarán disponibles para resolver dudas o colgar rutinas o ejercicios que hagan ellos, por supuesto los propios usuarios también tienen un apartado donde pueden personalizar sus propias tablas y compartirlas con el resto de usuarios.

La aplicación se hará en Android porque hay un estudio que revela que el mayor número de usuarios de Smartphone tienen como Sistema Operativo Android:



1.2 Objetivos del proyecto

Los objetivos a conseguir en este proyecto son los siguientes:

- Elaborar un sistema funcional de registro de usuarios conectado a una base de datos.
- Una base de datos donde se van a almacenar los datos de los usuarios que utilizará la aplicación.
- Usar la base de datos para que los usuarios puedan personalizar su tabla
- Crear una especie de foro donde los distintos usuarios puedan comunicarse entre sí y puedan pasar contenido.
- Elaborar un sistema de puntos donde se puedan puntuar las tablas que suben los usuarios para que a final del mes valorar la tabla con más puntuación y dicho usuario recibirá alguna bonificación o premio.
- Establecer una tienda donde se puedan comprar productos deportivos

2. ANÁLISIS

2.1 Acerca de la organización

Somos MyPower una empresa dedicada al ámbito deportivo, nos dedicamos a fabricar ropa de nuestra marca deportiva MP, así como accesorios deportivos como bandas, botellas deportivas (Shakers)…

Hemos decidido sacar una aplicación para móviles que nos permita hacer ejercicio de una forma planificada y estructurada mediante tablas que diseñan y programan nuestros profesionales, disponibles para todo tipo de usuarios.

Contamos con empleados profesionales en su campo desde programadores que nos ayudaran a diseñar y programar nuestra App a deportistas de elite que diseñaran tablas personalizadas y darán consejos, así como dietistas o nutricionistas que te recomendaran dietas o suplementos adecuados para tu objetivo.

- **Programadores:** 4
- **Deportistas de alto rendimiento:** 2
- **Nutricionistas:** 2

2.2 Análisis de requerimientos de la aplicación

Los requerimientos o requisitos funcionales que tiene que cumplir la aplicación son los siguientes:

Requerimientos Funcionales 1: Registro

1. La aplicación debe permitir al usuario introducir sus datos en el formulario de registro.
2. El sistema se encargará de validar los datos.
3. El sistema mostrará un mensaje de error si alguno de los datos es incorrecto o no cumple las condiciones especificadas del formulario de registro.
4. En el caso de que la validación sea correcta, el sistema se encargará de guardar los datos del usuario en base de datos
5. La aplicación mostrará un mensaje de bienvenida al usuario y le redirigirá a la pantalla principal con su sesión ya iniciada.

Requerimientos Funcionales 2: Identificación

1. Para iniciar sesión el usuario deberá identificarse con su nombre de usuario y contraseña. En el caso de ya haber iniciado sesión anteriormente el inicio de sesión será automático al iniciar la aplicación.
2. El sistema se encargará de validar y permitir (si los datos son correctos) o denegar (si los datos son incorrectos) el acceso a la aplicación.
3. El sistema mostrará un mensaje de error en el caso de que la validación sea incorrecta.
4. Si la validación es correcta, se mostrará un mensaje de bienvenida al usuario y le redirigirá a la pantalla principal

Requerimientos Funcionales 3: Cerrar Sesión

1. Todos los usuarios de la aplicación pueden finalizar sesión en la aplicación mediante el botón de "Cerrar de sesión".
2. Si el usuario pulsa el botón de cerrar sesión, el sistema mostrará un mensaje, para asegurarse de que el usuario quiere cerrar sesión, permitiendo aceptar o cancelar.
3. El usuario será redirigido a la pantalla principal de inicio de sesión.

Requerimientos Funcionales 4: Cuenta de usuario o Perfil

1. La aplicación deberá tener una pantalla en el sistema que permita al usuario consultar o modificar los datos de su cuenta.
2. Si el usuario modifica alguna información del perfil el sistema lo validará.
3. Si la validación es correcta, se actualizarán sus datos en la base de datos.

Requerimientos Funcionales 5: Pestaña Tablas

1. El usuario con la sesión iniciada podrá acceder a la pestaña ejercicios.
2. En esta pestaña podrá ver, eliminar o modificar sus tablas de ejercicios.
3. También podrá pedir una tabla personalizada en el botón de “Tablas Personalizada”
4. El usuario podrá añadir ejercicios a la tabla o eliminarlos, así como editar las series o repeticiones.

Requerimientos Funcionales 6: Pestaña Foro

1. El usuario podrá acceder a la pestaña foro de la aplicación.
2. En ella puede subir comentarios o dudas, tablas de ejercicios y puntuar otras tablas de otros usuarios

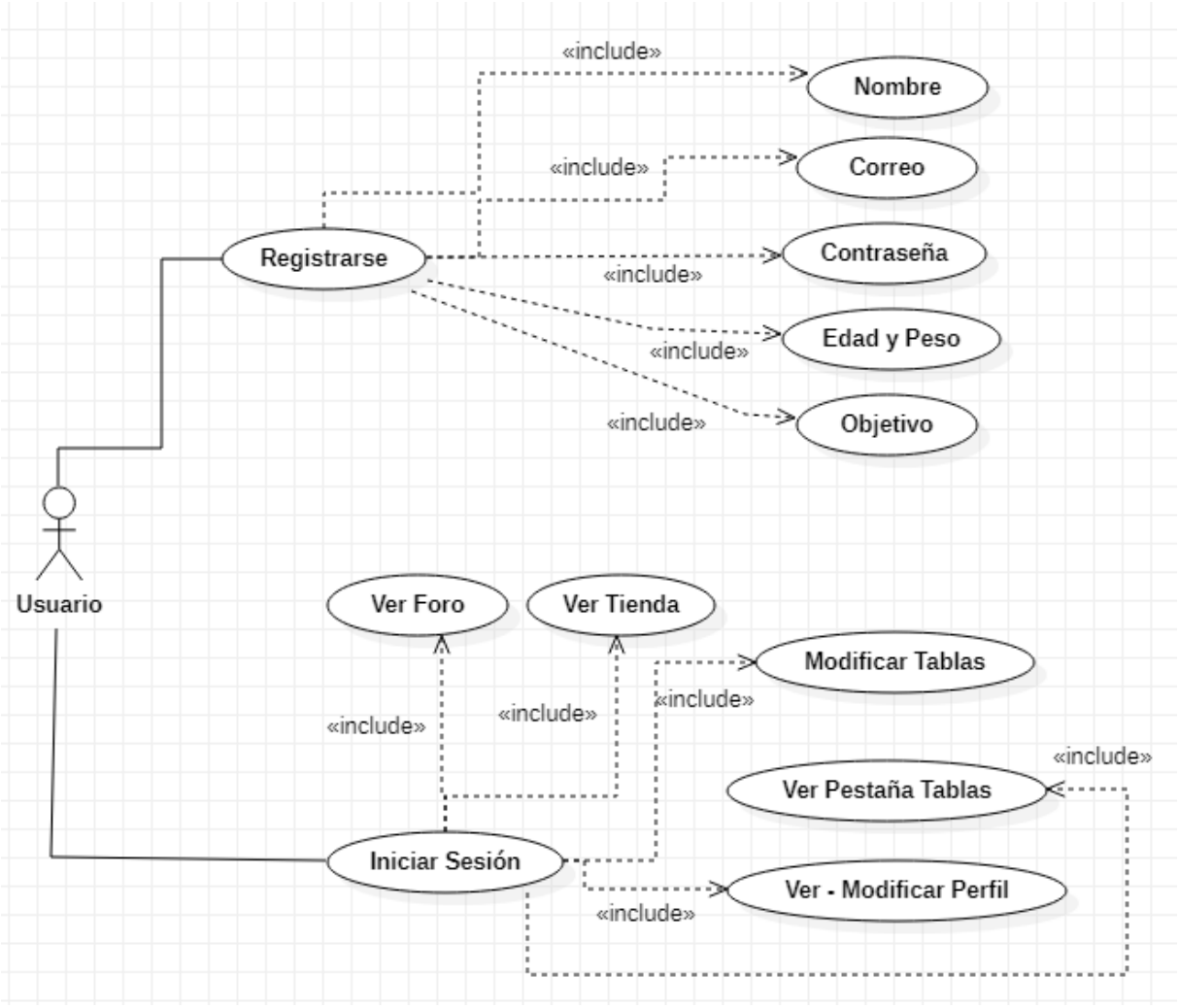
Requerimiento no Funcional 1: Tienda

1. Los usuarios podrán acceder a la tienda, para comprar accesorios de la marca de la empresa MyPower.

Requerimiento no Funcional 2: Premios en el Foro

1. Los usuarios con más puntuación en sus tablas en el foro tendrán la opción de ganar un paquete de merchandising de la empresa MyPower.

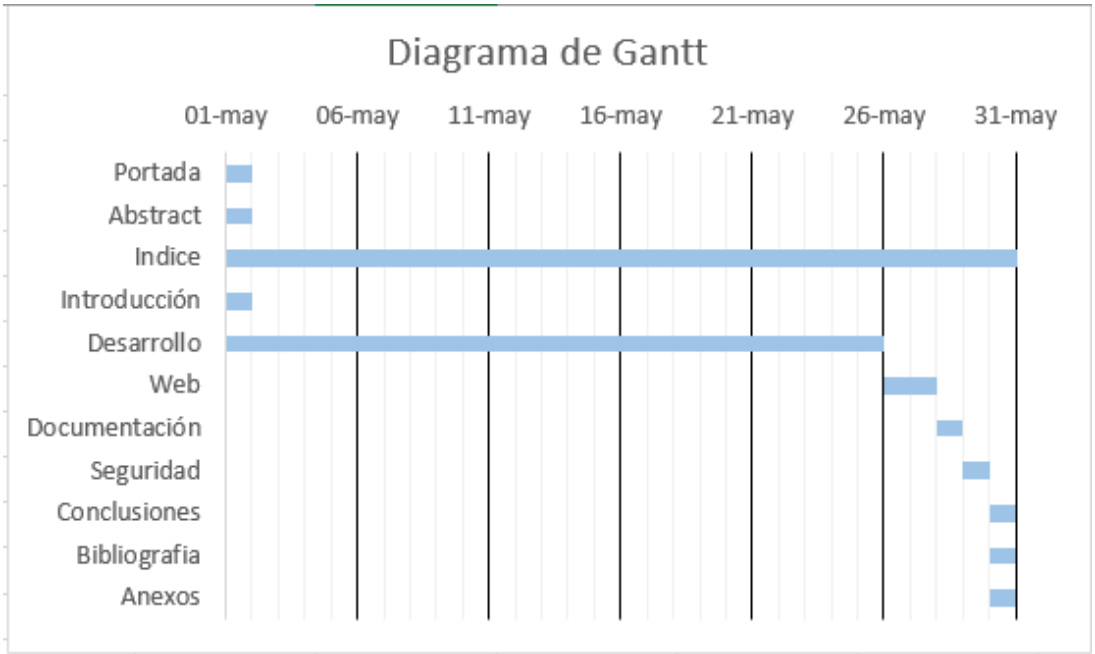
Este es el diagrama de casos de uso de lo que quiere el cliente que haga la aplicación:



2.3 Diagrama de Gantt

En este apartado vamos a ver el diagrama de Gantt, es decir, el ciclo de vida de las tareas del proyecto:

Número de Tarea	Fecha de inicio	Duración en días	Fecha fin
Portada	01-may	1	01-may
Abstract	01-may	1	01-may
Indice	01-may	31	30-may
Introducción	01-may	1	01-may
Desarrollo	01-may	25	26-may
Web	26-may	2	28-may
Documentación	28-may	1	29-may
Seguridad	29-may	1	30-may
Conclusiones	30-may	1	31-may
Bibliografia	30-may	1	31-may
Anexos	30-may	1	31-may



2.4 Requisitos

Al ser una aplicación desarrollada solo para Android y relacionada solo con el ámbito deportivo, no hacen falta ni muchos requisitos ni memoria para que la aplicación pueda correr en nuestro teléfono Android, está planificado que la aplicación pueda correr en el mayor número de dispositivos posibles:

- **Requiere Android:** Android 5.0 y versiones posteriores
- **Tamaño:** 15 MB o menos
- **Versión Actual:** 1.0
- **Clasificación de contenido:** PEGI 3
- **Desarrollador:** MyPower

2.5 Presupuesto

Teniendo en cuenta los empleados, los requerimientos y requisitos de la aplicación, el presupuesto calculando el Hardware y el software será el siguiente:

Red e Internet: Contrataremos una tarifa ilimitada para autónomos que cuenta entre otras cosas con 600 Mb de fibra de Vodafone por 57,84 €/mes.

Hardware: el hardware se compondrá de 4 ordenadores PcCom WorkStation, para los programadores, con suficiente potencia para este y futuros proyectos, precio 1014€.

Para los deportistas de alto rendimiento y nutricionistas se les dará un Smartphone Android a cada uno para que puedan entrar en la App y responder dudas en los foros, colgar sus ejercicios y todo lo anteriormente explicado, Xiaomi Redmi Note 8, 200€ cada uno.

Software: la mayoría del software utilizado será gratuito, pero hay que comprar las licencias Windows 10 con las que se va a trabajar, Windows 10 Pro 158€.

Uno de los ordenadores trabajará como servidor así que también habrá que adquirir una licencia Windows Server Essentials 2019 400€.

Salarios

- **Programadores:** 1.300€ = 5.200€
- **Nutricionistas:** 1.200€ = 2.400€
- **Deportistas de alto nivel:** 1.200€ = 2.400€

Con todo esto el proyecto sin contratiempos y en un mes calculo que el presupuesto debería de ser de unos 18.000€ el primer mes, aunque la suma de todo de 15.900€ aproximadamente, dejaremos un margen por si algo no sale bien.

2.6 Hardware, Sistemas Operativos y Aplicaciones

En este apartado vamos a tratar con el Hardware que se va a utilizar en la empresa, por qué hemos elegido Windows 10 Pro como sistema operativo y las Aplicaciones que vamos a utilizar para llevar a cabo el proyecto.

El Hardware con el que van a contar los 4 ordenadores PcCom es el siguiente:

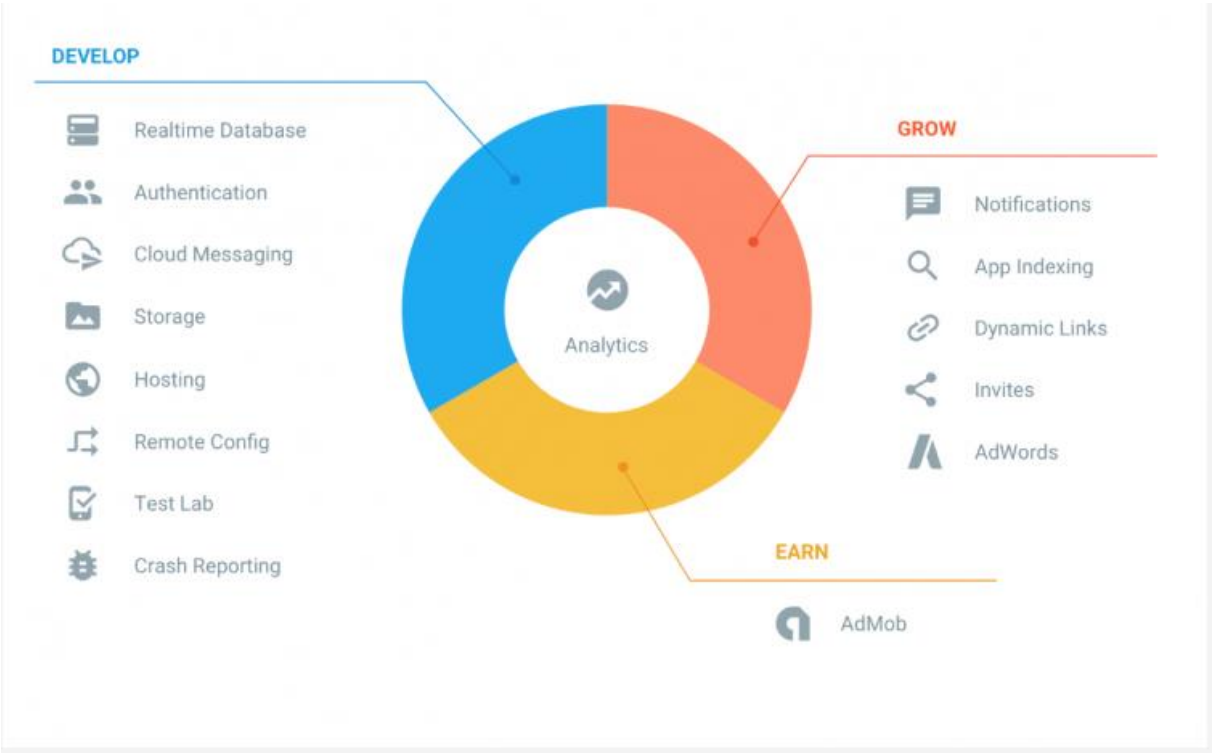
- **Fuente:** Corsair CV Series 650W 80 Plus Bronze
- **Placa:** MSI Z390-A PRO
- **Procesador:** Intel Core i5-9600K 3.7Ghz
- **Refrigeración:** Cooler Master Hyper 212X
- **Disco duro 1:** Toshiba OCZ TR200 SSD 240GB SATA3
- **Disco duro 2:** Seagate Barracuda 3.5” 1TB SATA3
- **Memoria RAM:** Kingston HyperX Fury Black 16GB DDR4 2400Mhz CL15 (2x8GB)
- **Tarjeta Gráfica:** Gigabyte GeForce GTX 1660 OC 6GB GDDR5
- **Caja/Torre:** Corsair Carbide 275Q USB 3.0 Negra

Hemos elegido Windows como sistema operativo porque es un sistema operativo fácil de usar, es muy intuitivo y sencillo de administrar, implementar o usar, es muy eficiente a la hora de realizar tareas y con transparencia a la hora de ejecutar procesos, es el más popular entonces es compatible con la mayoría de programas o dispositivos, tiene soporte técnico constante y es multi dispositivo.

El resto de Software utilizado es:

Android Studio: es un entorno de desarrollo integrado oficial para Android, lo anunció Google en la conferencia de Google I/O el 16 de mayo de 2013, Android Studio reemplazo a Eclipse como el IDE oficial para desarrollar aplicaciones de Android, está basado en IntelliJ y es gratuito, está disponible para MacOS, Windows y Linux.

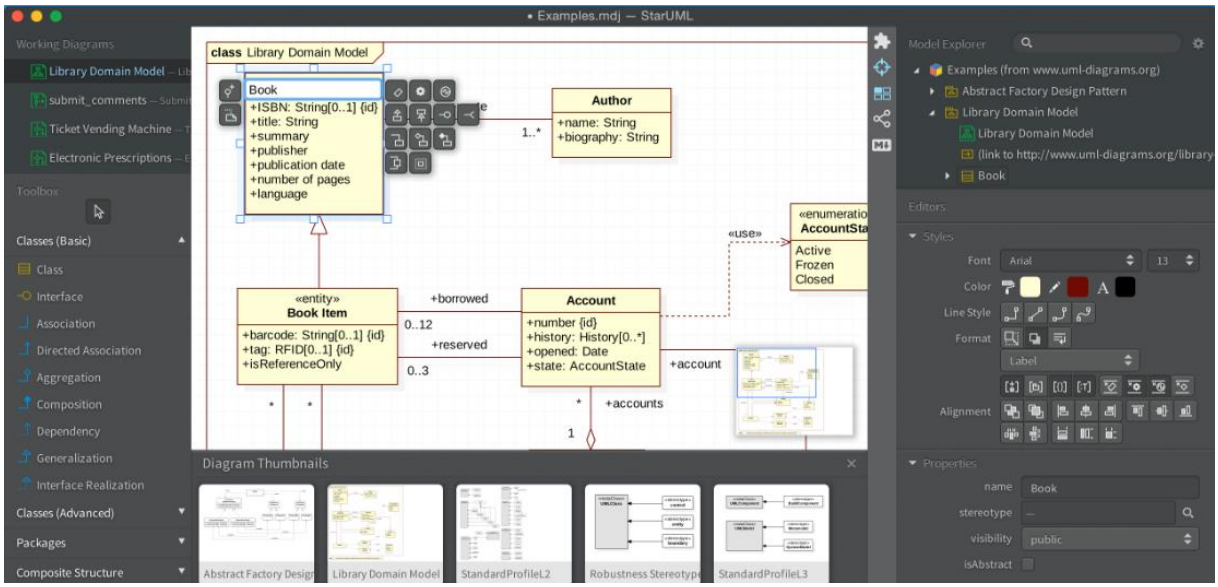
FireBase: firebase es una nueva y mejorada plataforma de desarrollo móvil en la nube de Google, sirve para diferentes plataformas como Android, IOS y web ¿Qué nos proporciona Firebase?



Firebase nos proporciona poder acceder a un servicio web para poder tener nuestra aplicación web trabajando con datos en la nube, es decir, es una API que guarda y sincroniza datos en la nube a tiempo real. Firebase tiene una versión gratuita y otra de pago:

Productos	Sin cargo Plan Spark Generous limits to get started	Pago por uso Plan Blaze Calcula los precios de las apps a gran escala. ✓ Se incluye el uso gratuito del plan Spark*
Pruebas A/B	Sin cargo	
Analytics	Sin cargo	
App Distribution	Sin cargo	
App Indexing	Sin cargo	
Authentication		
Autenticación telefónica: Canadá, EE.UU. y la India	10,000 por mes	USD 0.01 por verificación
Autenticación telefónica: Todos los demás países	10,000 por mes	USD 0.06 por verificación
Otros servicios de autenticación	✓	✓

StarUML: es un programa que nos va a ayudar a crear los diagramas de casos de usos de la aplicación, es una herramienta para el moldeamiento de los estándares UML, es un software libre. Una de las mejores aplicaciones gratis para trabajar UML.



3. BASE DE DATOS

En este apartado se van a tratar los aspectos de diseño de la base de datos del servidor. Los diagramas se realizarán como hemos explicado anteriormente con StarUML. Mostraremos el diagrama Entidad Relación, la transformación del diagrama Entidad – Relación al modelo Relacional y crearemos varios usuarios en la base de datos RealTime Database de Firebase.

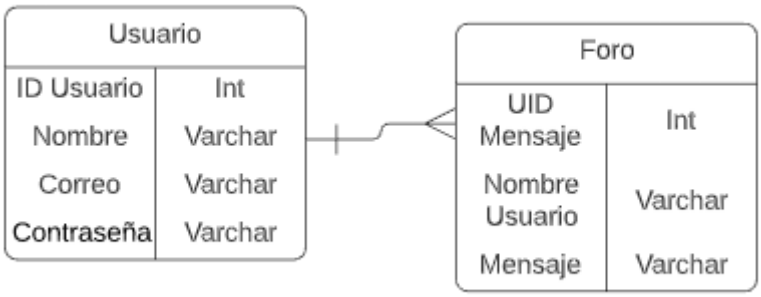
Para este proyecto hemos utilizado la Base de datos en la nube RealTime de Firebase, es una opción fácil y sencilla para implementar una base de datos a tu aplicación.

El cliente inicia las solicitudes, por lo que tiene un papel activo en la comunicación y espera la respuesta del servidor. El cliente interactúa con el usuario a través de la interfaz gráfica de la aplicación, el servidor RealTime espera las peticiones de los clientes interpretando un papel pasivo y tras la recepción procesa y envía los datos al cliente.

Para implementar la base de datos de Firebase, hay que sincronizar antes Firebase con Android Studio, hay que seguir las instrucciones que aparecen en la propia página de Firebase.

3.1 Diagrama Entidad – Relación

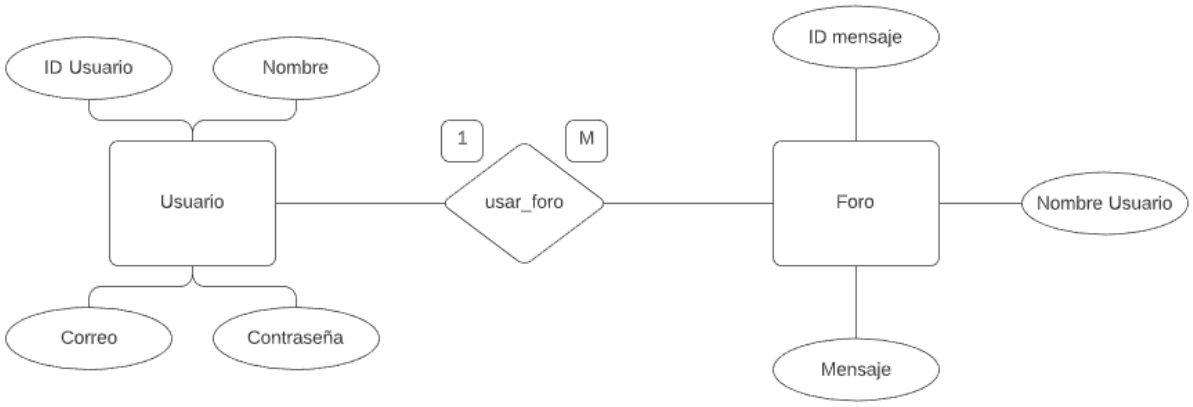
Un diagrama o modelo entidad-relación es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información, así como sus interrelaciones y propiedades.



Realizada con LucidChart

3.2 Diagrama Relacional

Es un modelo de organización y gestión de bases de datos consistente en el almacenamiento de datos en tablas compuestas por filas, o tuplas, y columnas o campos. Se distingue de otros modelos, como el jerárquico, por ser más comprensible para el usuario inexperto, y por basarse en la lógica de predicados para establecer relaciones entre distintos datos.



Realizado con LucidChart

4. DISEÑO DE LA APLICACIÓN

4.1 Diagramas de Casos de uso

Frontend y Backend

El proyecto al ser una aplicación deportiva, el usuario deberá registrarse, es decir, poner un correo, un nombre, una contraseña. Una vez registrado y con todos los datos el usuario podrá iniciar sesión y ver su información, perfil, el foro de la aplicación, la pestaña de tablas y la tienda. Los datos del usuario se guardarán en la Base de Datos.

Como ya he dicho anteriormente estamos utilizados Firebase de Google, para implementar distintos servicios en la nube a nuestra app, es decir, Firebase nos proporciona directamente el Backend ya listo para el desarrollo de aplicaciones web y apps para dispositivos. Podemos realizar un desarrollo acelerado de aplicaciones, ya que no necesitamos desarrollar la parte del servidor. Este tipo de servicios de computación en la nube se conoce como BaaS (Backend as a Service) en el que la tarea principal para el desarrollo backend será la configuración, en vez de la programación. El administrador podrá ver todos los usuarios a través de Firebase, ver sus correos, contraseñas y si es necesario reestablecer las contraseñas, eliminar alguna cuenta que no se comporte bien por el foro… También puede editar las tablas o rutinas, las dietas o los retos.

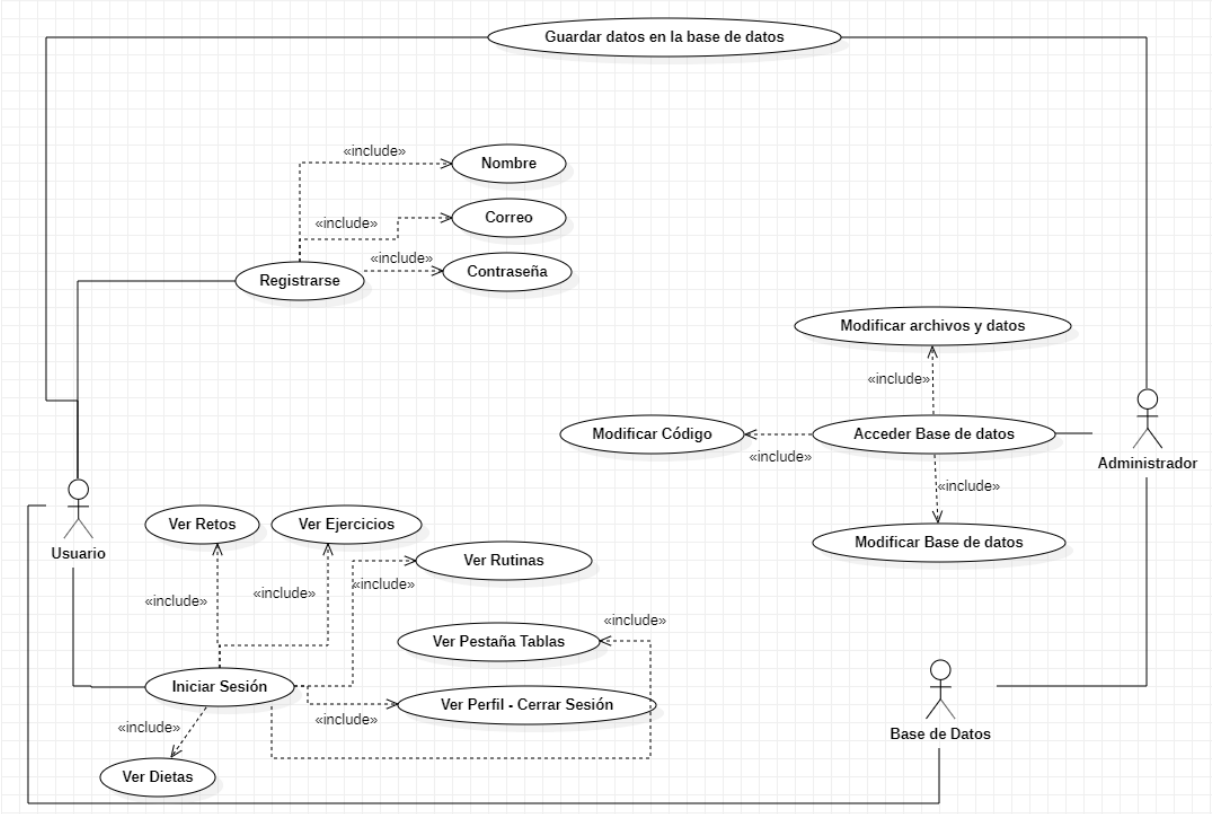


Diagrama de Clases

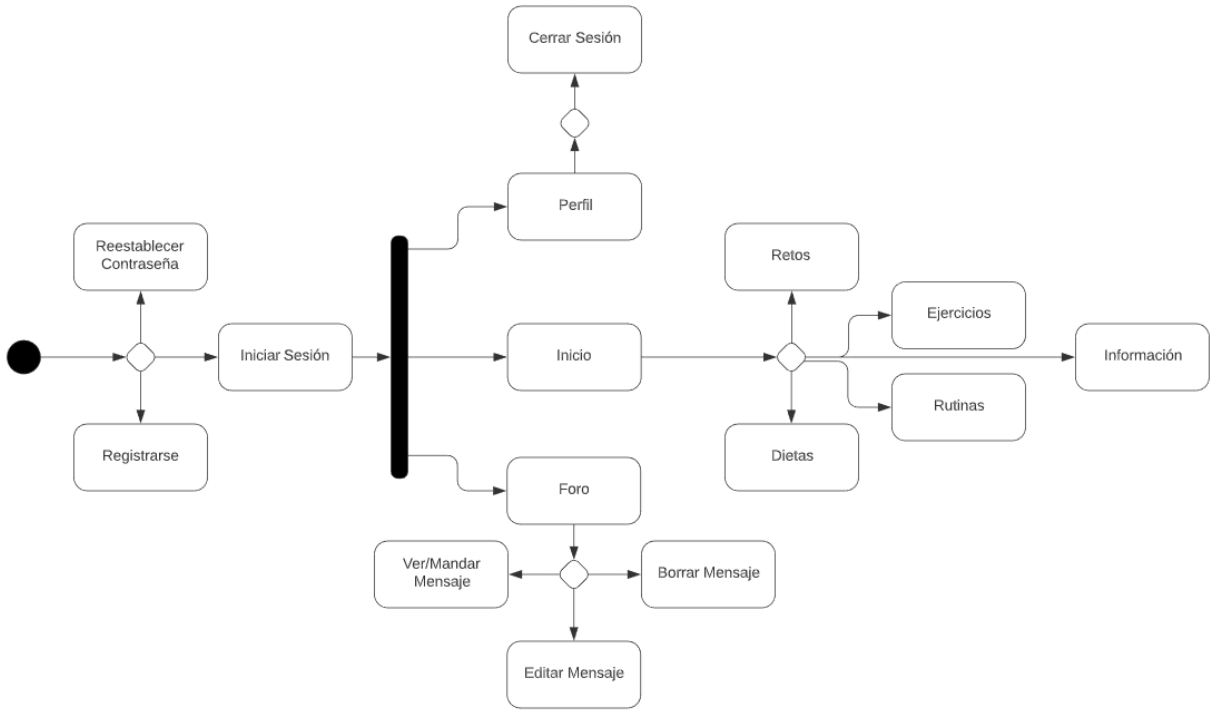
El diagrama de clases se ha hecho con un plugin de AndroidStudio llamado SimpleUML, puede encontrar en la tienda de plugins de esa misma aplicación, solo tenemos que ir a la carpeta de clases de las que queremos que nos genere el diagrama, darle click derecho y seleccionar “add to SimpleUML Diagram”, la aplicación nos generara los cuadros de clases con sus constructores y metodos, pero nosotros debemos agregar las implementaciones, extensiones o lo que creamos convenientes para nuestro proyecto.

En el indice hay un apartado llamado “ANEXO DIAGRAMA DE CLASES” pulsar para ver

Diagrama de estados

Un diagrama de estados, en ocasiones conocido como diagrama de máquina de estados, es un tipo de diagrama de comportamiento en el Lenguaje Unificado de Modelado (UML) que muestra transiciones entre diversos objetos. Para comprender los diferentes estados de un objeto, podrías visualizar todos los estados posibles y mostrar cómo un objeto llega a cada estado, y puedes hacerlo con un diagrama de estados UML.

Para hacer este diagrama hemos usado la página de es una página web que te permite hacer distintos tipos de diagramas.



5. PROGRAMACIÓN

Implementaciones hechas en el gradle (:app) necesarias para el proyecto:

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'androidx.appcompat:appcompat:1.0.2'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test:runner:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.1'

    //Implementaciones Firebase
    implementation 'com.google.firebase:firebase-analytics:17.2.2'
    implementation 'com.google.firebase:firebase-database:19.2.0'
    implementation 'com.google.firebase:firebase-auth:19.3.1'
    implementation 'com.google.firebase:firebase-storage:19.1.1'

    //Para diseñar el menu
    implementation 'com.google.android.material:material:1.0.0'


}


apply plugin: 'com.google.gms.google-services'
```

Permisos otorgados en el AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.fitwork">

    <uses-permission android:name="android.permission.INTERNET" />
```

MainActivity (Login) Variables:

```
public class MainActivity extends AppCompatActivity {

    EditText correo, contraseña;
    Button iniciarSesion, registrarUsuario, reestablecerContraseña;
    String email = " ";
    String password = " ";

    //Creamos el objeto Firebase para poder iniciar sesion
    FirebaseAuth mAuth;
```

MainActivity (Login) OnClick y métodos:

```
iniciarSesion.setOnClickListener((v) → {

        email = correo.getText().toString();
        password = contraseña.getText().toString();

        //Si no esta vacio hacemos el login
        if(!email.isEmpty() && !password.isEmpty()) {
            loginUser();

        }
        //Si no salta un mensaje de error
        else{
            Toast.makeText( context: MainActivity.this,  text: "Por favor completa los campos para iniciar sesión", Toast.LENGTH_SHORT).show();
        }
});
```

```
private void loginUser() {
    mAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener((task) → {
            //si la tarea fue exitosa se inicia sesion
            if (task.isSuccessful()) {
                //startActivity(new Intent(MainActivity.this, VentanaInicio.class));
                Intent intent = new Intent(getApplicationContext(), VentanaInicio.class);

                intent.putExtra( name: "string_usuario", email);

                startActivity(intent);
                //finish es para que no pueda volver a la pantalla anterior
                finish();
            }
            //Si no
            else {
                Toast.makeText( context: MainActivity.this,  text: "No se pudo iniciar la sesión, compruebe los datos", Toast.LENGTH_SHORT).show();
            }
    });
}
```

```
public void Registrarse(View view){
    Intent intent = new Intent( packageContext: MainActivity.this, VentanaRegistro.class);
    startActivity(intent);
}

public void Contraseña(View view) {
    Intent intent2 = new Intent( packageContext: MainActivity.this, ResetPassword.class);
    startActivity(intent2);
}

//Con este metodo lo que haremos es que reconozca si un usuario ha iniciado sesion y en vez de enviarlo a la pantalla para volver a iniciar sesion
//Aparecerá directamente en la ventana de incio
@Override
protected void onStart() {
    super.onStart();

    if (mAuth.getCurrentUser() !=null) {
        startActivity(new Intent( packageContext: MainActivity.this, VentanaInicio.class));
        finish();
    }
}
```

VentanaRegistro Variables y OnCreate:

```java
public class VentanaRegistro extends AppCompatActivity {


    EditText nombre, correo, contraseña;
    Button crearCuenta;

    //Variables de los datos que vamos a registrar
    String name = " ";
    String email = " ";
    String password = " ";

    //Objeto firebaseauth que nos permitira hacer la autenticacion con firebase
    private FirebaseAuth mAuth;
    //Objeto para la base de datos realtime de Firebase
    DatabaseReference mDatabase;
```

```java
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_ventana_registro);

    //Inicializamos FirebaseAuth
    mAuth = FirebaseAuth.getInstance();
    //Inicializamos la base de datos
    mDatabase = FirebaseDatabase.getInstance().getReference();

    nombre = findViewById(R.id.et_nombre);
    correo = findViewById(R.id.et_correo);
    contraseña = findViewById(R.id.et_contrasena);
    crearCuenta = findViewById(R.id.bt_registrar);

    crearCuenta.setOnClickListener((v) -> {
            name = nombre.getText().toString();
            email = correo.getText().toString();
            password = contraseña.getText().toString();

            if (!name.isEmpty() && !email.isEmpty() && !password.isEmpty()) {

                if(password.length() >= 6) {
                    //Si la contraseña tiene 6 o mas caracteres llamamos al metodo registrar usuario
                    registerUser();
                }
                else {
                    Toast.makeText( context: VentanaRegistro.this,  text: "La contraseña debe de tener al menos 6 carácteres", Toast.LENGTH_LONG).show();
                }

            }

            else {
                //Si algun campo se deja vacio,saldra este mensaje
                Toast.makeText( context: VentanaRegistro.this,  text: "Debe completar todos los campos", Toast.LENGTH_LONG).show();
            }
    });
```

VentanaRegistro Métodos:

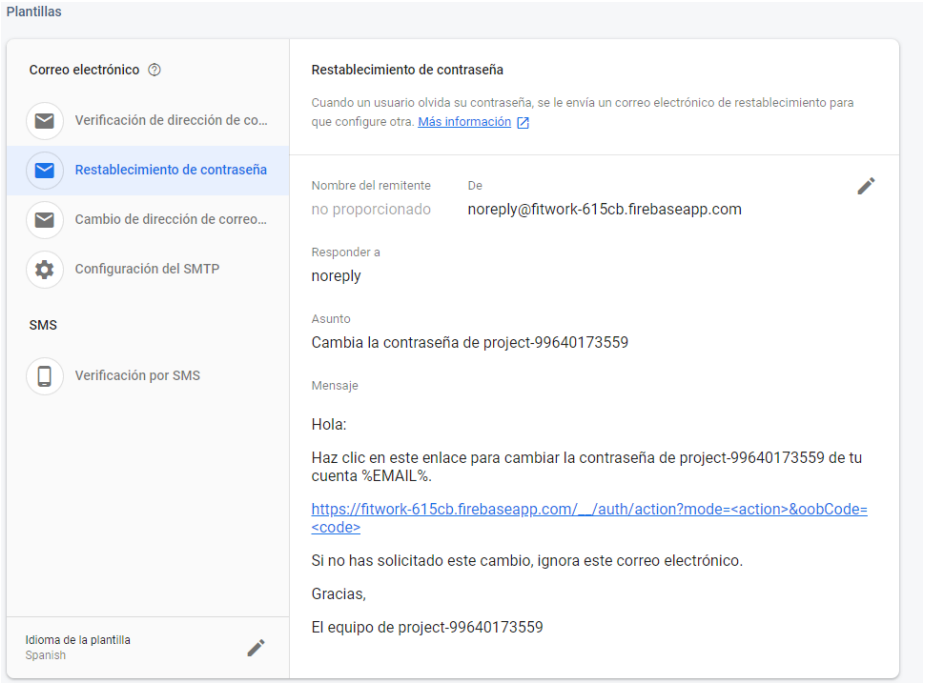
```
private void registerUser() {
    mAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener((task) → {

        //si la tarea se completo exitosamente, crearemos los datos en la base de datos
        if (task.isSuccessful()) {

            Map<String, Object> map = new HashMap<>();
            map.put("name", name);
            map.put("email", email);
            map.put("password", password);

            //Obtenemos el usuario que se va a registrar
            String id = mAuth.getCurrentUser().getUid();

            mDatabase.child("Users").child(id).setValue(map).addOnCompleteListener((task2) → {
                    if (task2.isSuccessful()) {
                        //si los usuarios se crean exitosamente los enviamos a otra pantalla
                        startActivity(new Intent( packageContext: VentanaRegistro.this, VentanaPerfil.class));
                        finish();
                    }
                    else {
                        Toast.makeText( context: VentanaRegistro.this,  text: "No se pudieron crear los datos correctamente", Toast.LENGTH_LONG).show();
                    }
                });
            }
            else {
                //Si no enviaremos un mensaje al usuario de que no se ha podido guardar los datos
                Toast.makeText( context: VentanaRegistro.this,  text: "No se pudo registrar este usuario", Toast.LENGTH_LONG).show();
            }
    });
}
```

ResetPassword Variables y OnCreate:

```
public class ResetPassword extends AppCompatActivity {

    EditText correo;
    Button reestablecer;
    String email = " ";

    FirebaseAuth mAuth;

    //objeto de un loader para mostrar el envio del correo electronico
    ProgressDialog mDialog;
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_reset_password);

    mAuth = FirebaseAuth.getInstance();
    mDialog = new ProgressDialog( context: this);

    correo = findViewById(R.id.et_correoContraseña);
    reestablecer = findViewById(R.id.bt_reestablecerPassword);

    reestablecer.setOnClickListener((v) → {
            //obtenemos el email
            email = correo.getText().toString();

            //antes de ejecutar el metodo vamos a validar si el correo esta puesto
            if (!email.isEmpty()) {
                //mostraremos un mensaje para el usuario
                mDialog.setMessage("Espere unos segundos...");
                //Asi el usuario no podra quitar el mensaje hasta que se termine
                mDialog.setCanceledOnTouchOutside(false);
                //se muestra
                mDialog.show();
                resetPassword();
            }
            else {
                Toast.makeText( context: ResetPassword.this,  text: "Debe ingresar el email", Toast.LENGTH_SHORT).show();
            }
    });
}
```

Plantilla FireBase para poder reestablecer la contraseña:



VentanaInicio OnCreate:

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_ventana_inicio);

    //Declaramos e inicializamos el menu
    BottomNavigationView bottomNavigationView = findViewById(R.id.bottomNavigation);
    //Decimos el objeto que esta seleccionando
    bottomNavigationView.setSelectedItemId(R.id.home);

    //La acción que pasa depende del boton que seleccionemos
    bottomNavigationView.setOnNavigationItemSelectedListener((menuItem) -> {
            switch (menuItem.getItemId()) {

                case R.id.rutinas:
                    //startActivity(new Intent(getApplicationContext(), VentanaForo.class));
                    Intent recuperarUsuario = getIntent();

                    String string_usuario_recibido2 = recuperarUsuario.getStringExtra( name: "string_usuario");

                    Intent intentNombreForo = new Intent(getApplicationContext(), VentanaForo.class);

                    intentNombreForo.putExtra( name: "string_usuario", string_usuario_recibido2);

                    startActivity(intentNombreForo);
                    overridePendingTransition( enterAnim: 0, exitAnim: 0);
                    return true;

                case R.id.home:
                    return true;

                case R.id.cuenta:
                    startActivity(new Intent(getApplicationContext(), VentanaPerfil.class));
                    overridePendingTransition( enterAnim: 0, exitAnim: 0);
                    return true;

            }
            return false;
    });
```

```
        guia = findViewById(R.id.bt_guiaEjercicios);
        dietas = findViewById(R.id.bt_dietas);
        retos = findViewById(R.id.bt_retos);
        rutinas = findViewById(R.id.bt_rutinas);
        informacion = findViewById(R.id.bt_informacion);

        guia.setOnClickListener((v) -> {
                Intent intentGuia = new Intent( packageContext: VentanaInicio.this, VentanaGuia.class);
                startActivity(intentGuia);
        });

        dietas.setOnClickListener((v) -> {
                Intent intentDietas = new Intent( packageContext: VentanaInicio.this, VentanaDietas.class);
                startActivity(intentDietas);
        });

        retos.setOnClickListener((v) -> {
                Intent intentRetos = new Intent( packageContext: VentanaInicio.this, VentanaRetos.class);
                startActivity(intentRetos);
        });

        rutinas.setOnClickListener((v) -> {
                Intent intentRutinas = new Intent( packageContext: VentanaInicio.this, VentanaRutinas.class);
                startActivity(intentRutinas);
        });

        informacion.setOnClickListener((v) -> {
                Intent intentInformacion = new Intent( packageContext: VentanaInicio.this, VentanaInformacion.class);
                startActivity(intentInformacion);
        });
    }
}
```

VentanaForo Varibles y OnCreate:

```
public class VentanaForo extends AppCompatActivity {

    private List<Foro> listMensajes = new ArrayList<~>();
    ArrayAdapter<Foro> arrayAdapterEjercicio;

    EditText menU;
    ListView listV_mensajes;
    String nombre, mensaje;

    FirebaseAuth mAuth;
    FirebaseDatabase firebaseDatabase;
    DatabaseReference databaseReference;

    Foro mensajeSelected;
```

```
setContentView(R.layout.activity_ventana_foro);
BottomNavigationView bottomNavigationView = findViewById(R.id.bottomNavigation);
bottomNavigationView.setSelectedItemId(R.id.rutinas);
bottomNavigationView.setOnNavigationItemSelectedListener((menuItem) → {
        switch (menuItem.getItemId()) {

            case R.id.rutinas:
                return true;

            case R.id.home:
                startActivity(new Intent(getApplicationContext(), VentanaInicio.class));
                overridePendingTransition( enterAnim: 0, exitAnim: 0);
                return true;

            case R.id.cuenta:
                startActivity(new Intent(getApplicationContext(), VentanaPerfil.class));
                overridePendingTransition( enterAnim: 0, exitAnim: 0);
                return true;

        }
        return false;
});

mAuth = FirebaseAuth.getInstance();

menU = findViewById(R.id.mensaje);
listV_mensajes = findViewById(R.id.datos);

//Metodo para inicializar Firebase
inicializarFirebase();
listarDatos();

listV_mensajes.setOnItemClickListener((parent, view, position, id) → {
        mensajeSelected = (Foro) parent.getItemAtPosition(position);

        menU.setText(mensajeSelected.getMensaje());
```

VentanaForo Métodos:

```
private void listarDatos() {
    databaseReference.child("Foro").addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            listMensajes.clear();

            //recorremos el objeto snapshot y obtenemos el hijo, con el objeto Foro obtenemos la clase y la guardamos y finalmente a la lista le añadimos el objeto
            for (DataSnapshot objSnapshot : dataSnapshot.getChildren()) {
                Foro j = objSnapshot.getValue(Foro.class);
                listMensajes.add(j);

                //Creamos el arrayAdapter y le colocamos una plantilla a la lista listEjercicio, finalmente añadimos al listView el nuevo adapter
                arrayAdapterEjercicio = new ArrayAdapter<Foro>( context: VentanaForo.this, android.R.layout.simple_list_item_1, listMensajes);
                listV_mensajes.setAdapter(arrayAdapterEjercicio);
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {

        }
    });
}
```

```java
private void inicializarFirebase() {
    FirebaseApp.initializeApp(this);
    firebaseDatabase = FirebaseDatabase.getInstance();
    //firebaseDatabase.setPersistenceEnabled(true);
    databaseReference = firebaseDatabase.getReference();
    //mAuth = FirebaseAuth.getInstance();
}

//Agregamos el menu a la ventana rutina de los tres botones a la derecha
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_rutina, menu);
    return super.onCreateOptionsMenu(menu);

}
```

Boton Añadir Mensaje:

```java
public boolean onOptionsItemSelected(MenuItem item) {

    mensaje = menU.getText().toString();
    switch (item.getItemId()) {

        case R.id.add:{
            if (mensaje.isEmpty()) {
                validacion();
            }
            else {
                //Objeto ejercicio creado en model
                Foro j = new Foro();
                j.setUid(UUID.randomUUID().toString());

                Intent intent = getIntent();

                String nombre = intent.getStringExtra( name: "string_usuario");

                j.setNombre(nombre);
                j.setMensaje(mensaje);

                databaseReference.child("Foro").child(j.getUid()).setValue(j);

                Toast.makeText( context: this,  text: "Se ha añadido Correctamente", Toast.LENGTH_SHORT).show();
                limpiarCajas();
            }
            break;
        }
```

Botón Editar/Guardar Mensaje:

```
case R.id.save:{
    //hacemos que cuando seleccionemos un item de la lista nos rellene los editText con sus datos,
    //podemos modificarlos y cuando le demos al boton de guardar, se actualizaran los datos
    Foro j = new Foro();
    j.setUid(mensajeSelected.getUid());

    j.setMensaje(menU.getText().toString().trim());

    databaseReference.child("Foro").child(j.getUid()).setValue(j);

    Toast.makeText( context: this,  text: "Se ha editado Correctamente", Toast.LENGTH_SHORT).show();
    limpiarCajas();
    break;
}
```

Botón Eliminar Mensaje:

```
        case R.id.delete:{

            if (mensaje.isEmpty()) {
                validacion2();
            }
            else {
                Foro j = new Foro();
                j.setUid(mensajeSelected.getUid());
                databaseReference.child("Foro").child(j.getUid()).removeValue();

                Toast.makeText( context: this,  text: "Se ha borrado correctamente", Toast.LENGTH_SHORT).show();
                limpiarCajas();
            }
            break;
        }
        default:break;
    }
    return true;
}
```

Métodos Validaciones y limpiar el texto:

```
private void validacion2() {
    Toast.makeText( context: this,  text: "Debe seleccionar un objeto de la lista", Toast.LENGTH_SHORT).show();
}

private void validacion() {

    mensaje = menU.getText().toString();

    if (mensaje.isEmpty()) {
        menU.setError("Campo Requerido");
    }
}

private void limpiarCajas() { menU.setText(""); }
```

VentanaForo objeto Foro (sirve para subir el mensaje a Firebase):

```java
public class Foro {

    private String uid;
    private String nombre;
    private String mensaje;

    public Foro() {
    }

    public String getUid() { return uid; }

    public void setUid(String uid) { this.uid = uid; }

    public String getNombre() { return nombre; }

    public void setNombre(String nombre) { this.nombre = nombre; }

    public String getMensaje() { return mensaje; }

    public void setMensaje(String mensaje) { this.mensaje = mensaje; }

    @Override
    public String toString() {

        return  nombre + "\n " + mensaje;
    }
```

VentanaPerfil Variables y OnCreate:

```
public class VentanaPerfil extends AppCompatActivity {

    Button CerrarSesion, SubirFoto;
    TextView nombre, correo;

    //Creamos el objeto Firebase para poder cerrar la sesion
    FirebaseAuth mAuth;
    //Objeto DataBase para recoger datos de la base de datos de FireBase
    DatabaseReference mDataBase;
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_ventana_perfil);

    //inicializamos el objeto
    mAuth = FirebaseAuth.getInstance();
    mDataBase = FirebaseDatabase.getInstance().getReference();

    nombre = findViewById(R.id.tv_nombre);
    correo = findViewById(R.id.tv_correo);

    CerrarSesion = findViewById(R.id.bt_CerrarSesion);

    CerrarSesion.setOnClickListener((v) -> {
            //Cerramos sesion cuando le damos al boton y volvemos a la pagina de inicio de sesion
            mAuth.signOut();
            startActivity(new Intent( packageContext: VentanaPerfil.this, MainActivity.class));
            finish();
    });
```

```
//Llamamos al metodo con el que vamos a recoger y poner los datos de los usuarios
getUserInfo();

BottomNavigationView bottomNavigationView = findViewById(R.id.bottomNavigation);

bottomNavigationView.setSelectedItemId(R.id.cuenta);

bottomNavigationView.setOnNavigationItemSelectedListener((menuItem) -> {
        switch (menuItem.getItemId()) {

            case R.id.rutinas:
                startActivity(new Intent(getApplicationContext(), VentanaForo.class));
                overridePendingTransition( enterAnim: 0, exitAnim: 0);
                return true;

            case R.id.home:
                startActivity(new Intent(getApplicationContext(), VentanaInicio.class));
                overridePendingTransition( enterAnim: 0, exitAnim: 0);
                return true;

            case R.id.cuenta:
                return true;

        }
        return false;
});
```

VentanaPerfil Métodos:

```
//Con este metodo vamos a acceder al nodo hijo Users para obtener el id del usuario con el que hemos iniciado sesion
private void getUserInfo() {
    //Obtenemos el id
    String id = mAuth.getCurrentUser().getUid();
    mDataBase.child("Users").child(id).addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            //Decimos que si ese nodo existe, recogemos los datos
            if (dataSnapshot.exists()) {
                String name = dataSnapshot.child("name").getValue().toString();
                String email = dataSnapshot.child("email").getValue().toString();

                //Y ponemos los datos recogidos
                nombre.setText(name);
                correo.setText(email);
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {

        }
    });
}
```

De todos los botones de la pantalla de Inicio voy a enseñar el código de un botón porque son muy parecidos todos.

VentanaGuia Variables (Guia de ejercicios) para esto he tenido que utilizar un RecyclerView y crear un objeto para poder añadir los ítems a la lista (Métodos y OnCreate):

```
public class VentanaGuia extends AppCompatActivity {

    private RecyclerView rvMuscles;
    private GuiaAdapter adapter;
    private List<guiaEjercicios> listaMusculos;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ventana_guia);

        //metodos para crear el recyclerview
        initViews();
        llenarDatosMusculos();
        setupMuscleAdapter();

    }
```

VentanaGuia Métodos:

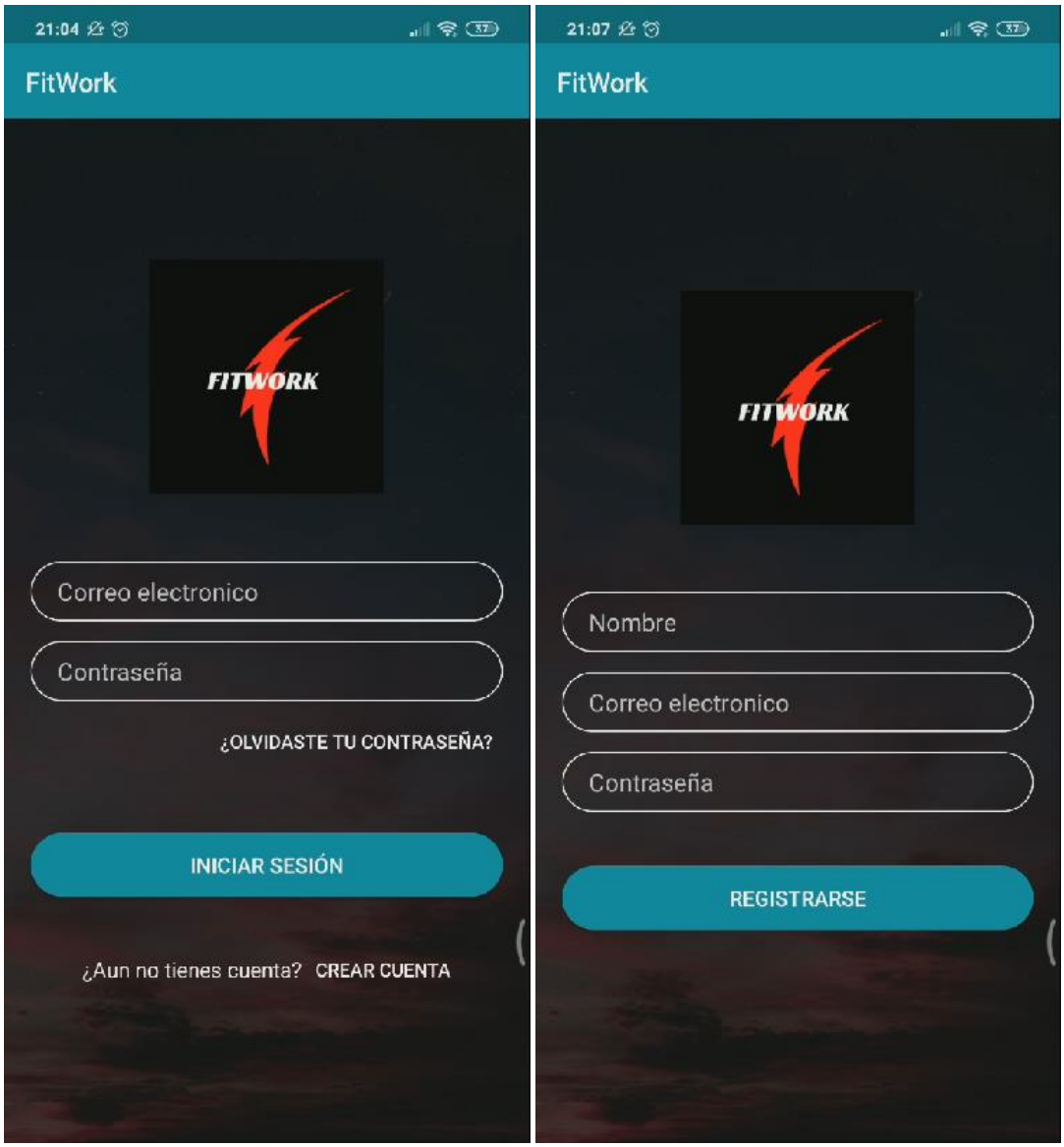
```
//lo usamos para inicializar y crear el adapter para meterlo dentro del recyclerview
private void setupMuscleAdapter() {

    adapter = new GuiaAdapter(listaMusculos);
    rvMuscles.setAdapter(adapter);

    adapter.setOnClickListener((view) -> {

            if (listaMusculos.get(rvMuscles.getChildAdapterPosition(view)).getNombre() == "Pectorales") {
                Intent intent = new Intent(getApplicationContext(), VentanaPectoral.class);
                startActivity(intent);
            }

            else if (listaMusculos.get(rvMuscles.getChildAdapterPosition(view)).getNombre() == "Espalda") {
                Intent intent = new Intent(getApplicationContext(), VentanaEspalda.class);
                startActivity(intent);
            }

            else if (listaMusculos.get(rvMuscles.getChildAdapterPosition(view)).getNombre() == "Biceps") {
                Intent intent = new Intent(getApplicationContext(), VentanaBiceps.class);
                startActivity(intent);
            }

            else if (listaMusculos.get(rvMuscles.getChildAdapterPosition(view)).getNombre() == "Triceps") {
                Intent intent = new Intent(getApplicationContext(), VentanaTriceps.class);
                startActivity(intent);
            }

            else if (listaMusculos.get(rvMuscles.getChildAdapterPosition(view)).getNombre() == "Hombros") {
                Intent intent = new Intent(getApplicationContext(), VentanaHombros.class);
                startActivity(intent);
            }

            else if (listaMusculos.get(rvMuscles.getChildAdapterPosition(view)).getNombre() == "Piernas") {
                Intent intent = new Intent(getApplicationContext(), VentanaPiernas.class);
                startActivity(intent);
```

```
//Rellenamos el recyclerview con los parametros pasados en el metodo guiaholder de la clase GuiaAdapter
private void llenarDatosMusculos() {

    //creamos la lista y la llenamos
    listaMusculos = new ArrayList<>();
    listaMusculos.add(new guiaEjercicios( nombre: "Pectorales",  info: "Ejercicios para pecho", R.drawable.pecho));
    listaMusculos.add(new guiaEjercicios( nombre: "Espalda",  info: "Ejercicios para espalda y dorsal",R.drawable.espalda));
    listaMusculos.add(new guiaEjercicios( nombre: "Biceps",  info: "Ejercicios para biceps y antebrazo",R.drawable.biceps));
    listaMusculos.add(new guiaEjercicios( nombre: "Triceps",  info: "Ejercicios para triceps",R.drawable.triceps));
    listaMusculos.add(new guiaEjercicios( nombre: "Hombros",  info: "Ejercicios para hombros",R.drawable.hombros));
    listaMusculos.add(new guiaEjercicios( nombre: "Piernas",  info: "Ejercicios para piernas y gluteos",R.drawable.piernas));
    listaMusculos.add(new guiaEjercicios( nombre: "Abdomen",  info: "Ejercicios para abdomen",R.drawable.abdomen));


}
```

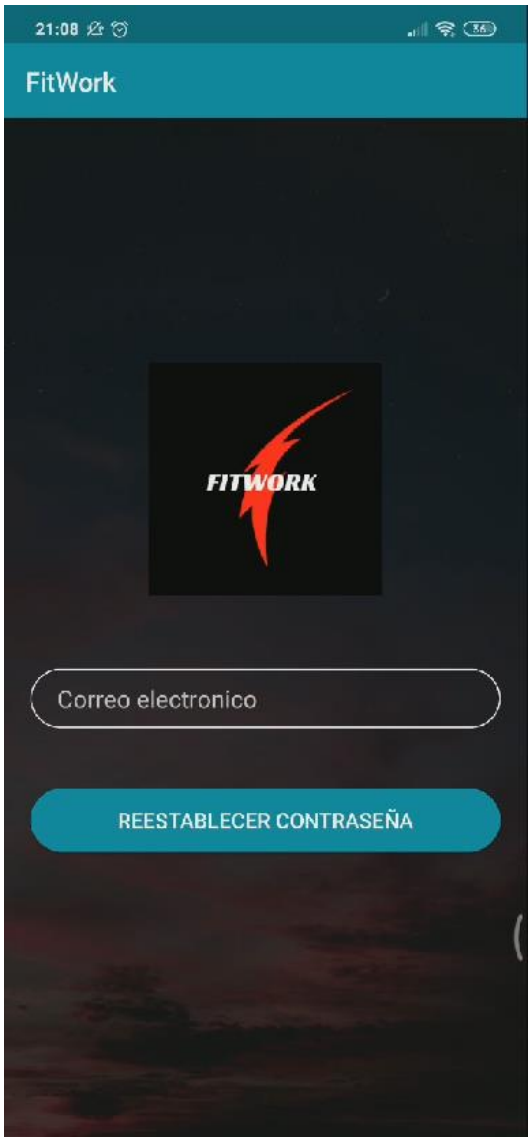
```
private void initViews() {
    //Asignamos al recyclerview un linearLayout y le decimos que su tamaño sea fijo
    rvMuscles = findViewById(R.id.rv_muscles);
    rvMuscles.setLayoutManager(new LinearLayoutManager( context: this));
    rvMuscles.setHasFixedSize(true);


}
```

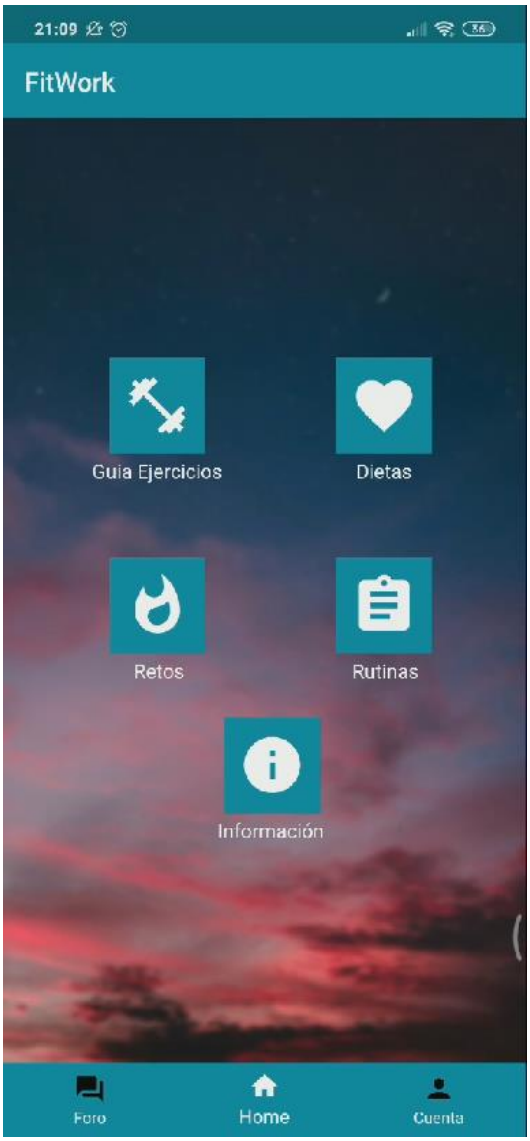
6. DESARROLLO DE INTERFACES Y MANUAL DE USUARIO

La interfaz de la aplicación se ha diseñado para que sea una interfaz simple, atractiva y fácil de usar, las capturas de pantalla de la aplicación se están tomando con una aplicación que duplica o proyecta la pantalla del dispositivo SmartPhone llamada LestView, tiene que estar instalada tanto en el PC como en el SmartPhone.

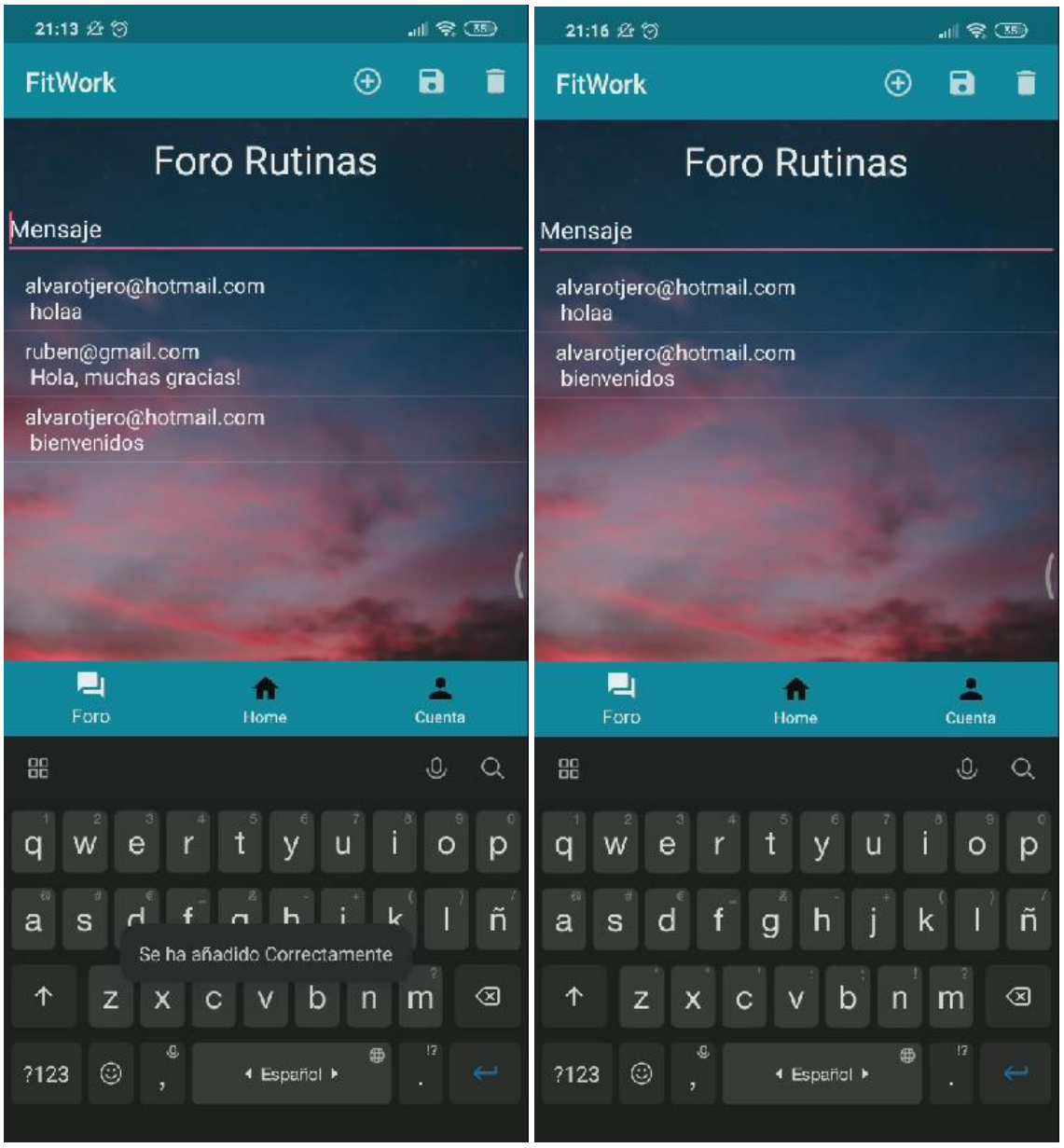


Aquí vamos a mostrar la pantalla Principal de la aplicación, esta compuesta de 5 botones y un menu situado abajo del todo que consta de tres botones de navegación:

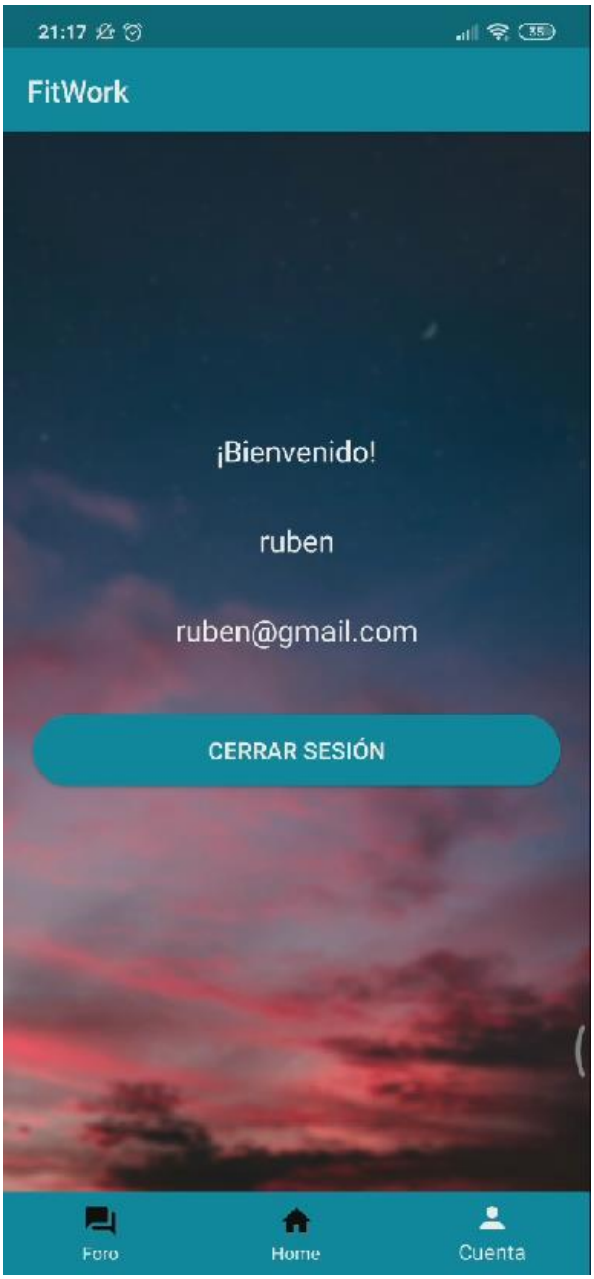




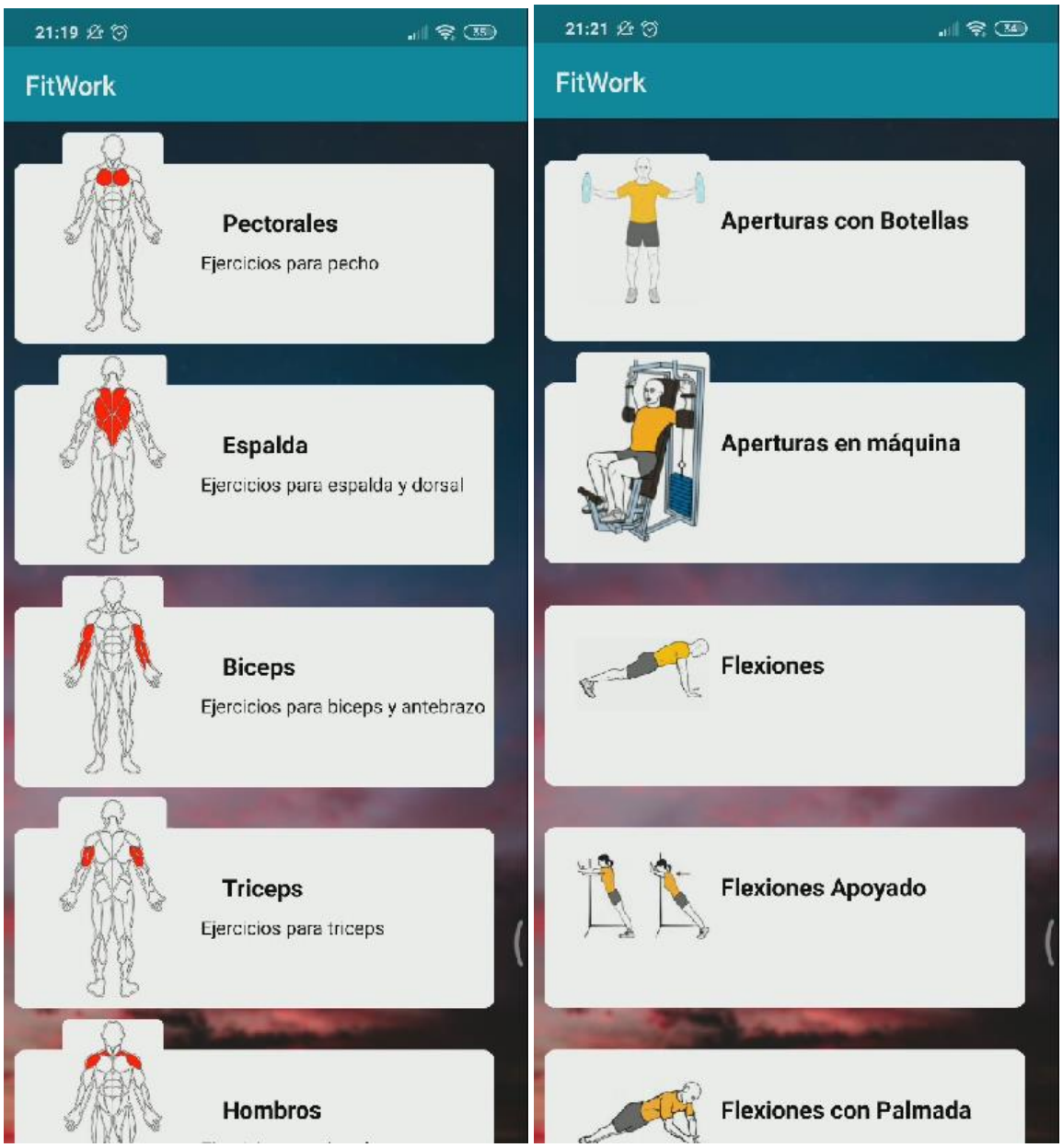
Vamos a mostrar el funcionamiento del foro, es un foto sencillo que tiene tres botones en la esquina superior derecha, uno para subir el mensaje, otro para editarlo y otro para eliminarlo, cuando escribimos y subimos el mensaje, este pone primero el correo del usuario que ha escrito ese mensaje y debajo escribe el mensaje, cuando seleccionamos el mesaje aparece escrito en la caja de mensaje y si lo editamos y presionamos el boton de guardar el mesaje queda editado y lo mismo con el de borrar, hay que seleccionar el mensaje y presionar el boton de la papelera:



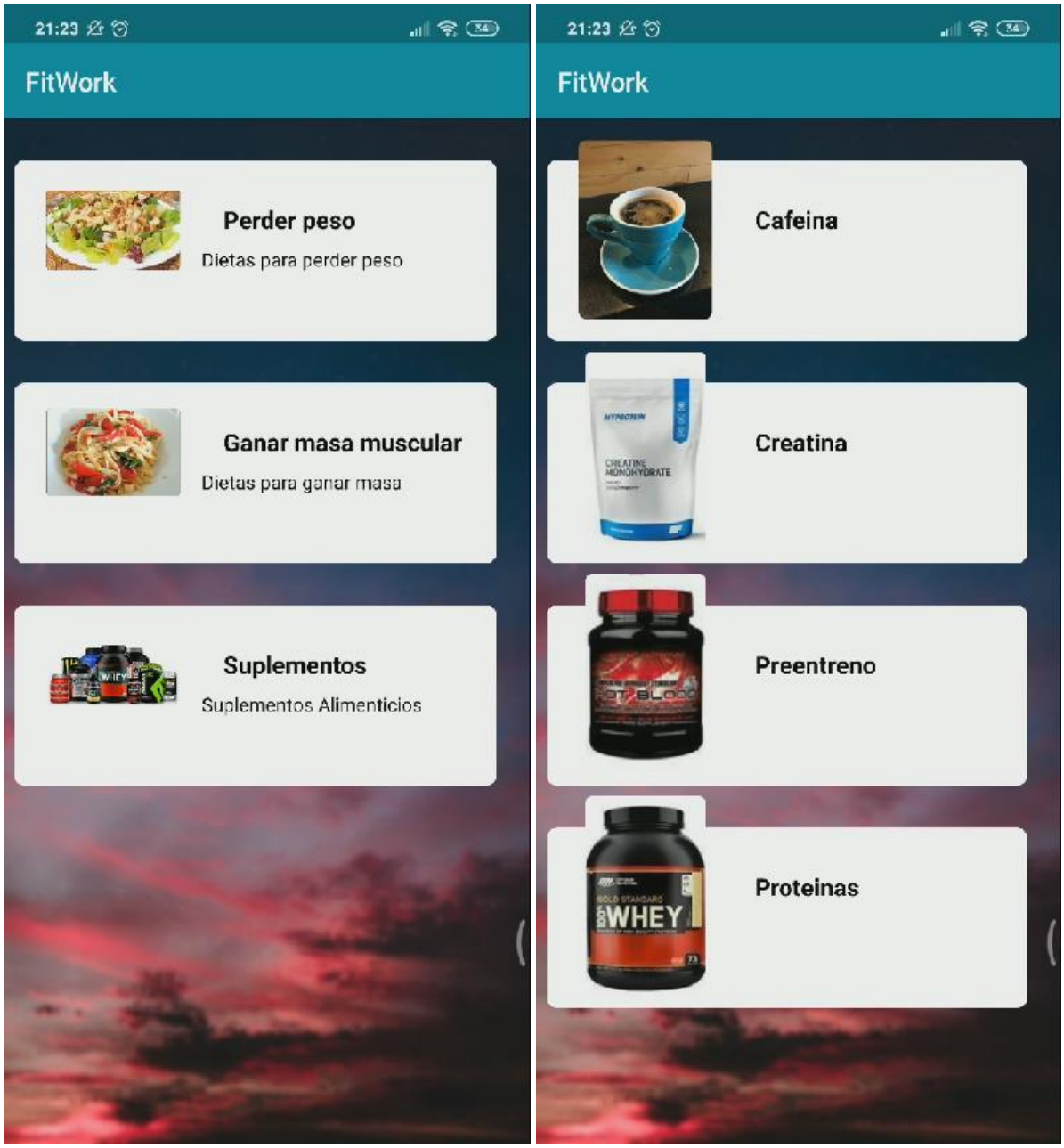
La ventana de perfil es muy sencilla, muestra un mensaje de bienvenida con los datos introducidos por el usuario junto con un boton de cerrar sesión:

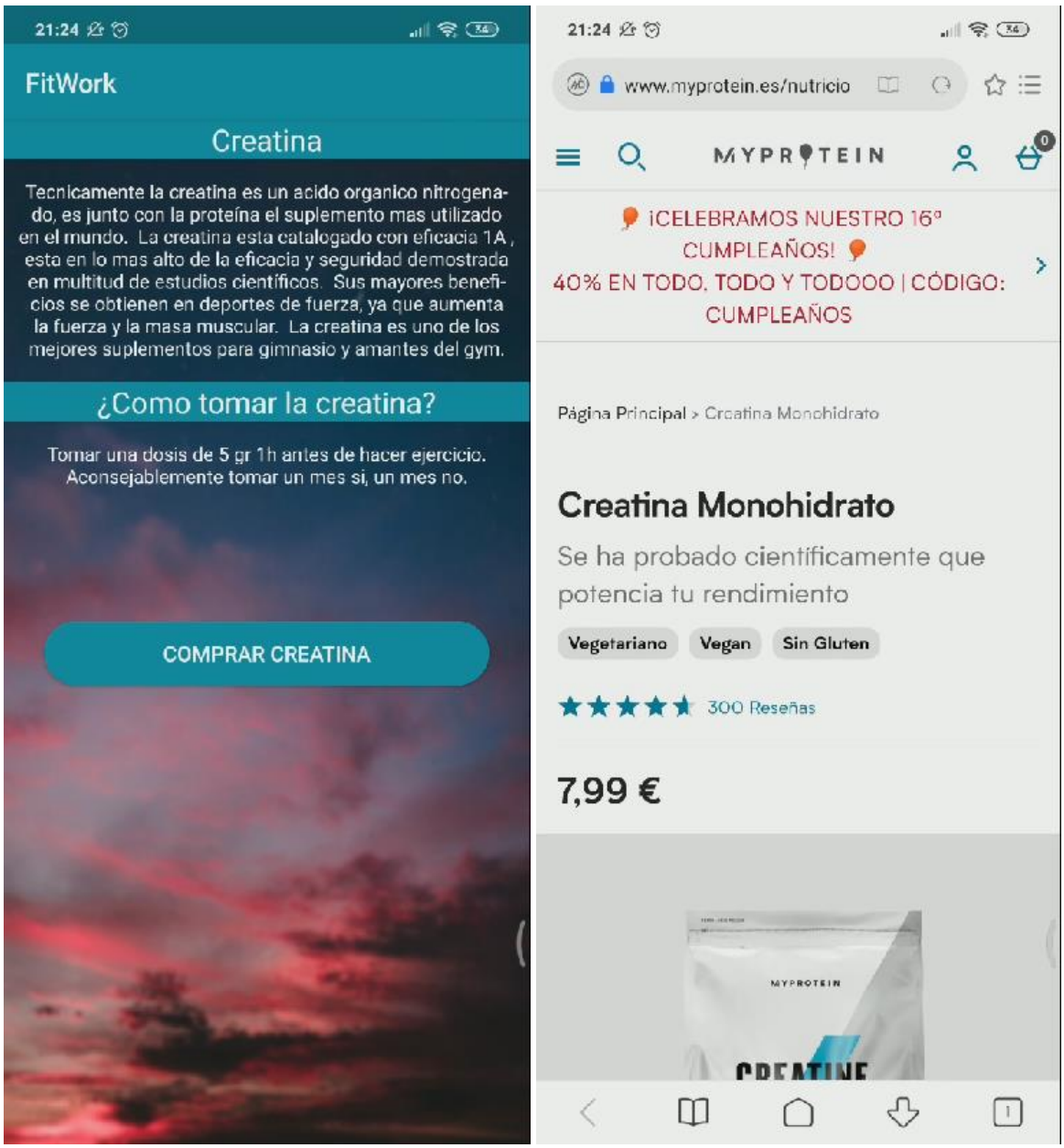


Dentro de los botones Guia Ejercicios, Dietas, Retos y Rutinas se encuentra un menú creado con un RecyclerView que muestran distintos elementos de una lista de objetos, por ejemplo, el botón Guia Ejercicios contiene distintas zonas del cuerpo que se quieran entrenar, dentro de cada zona muscular encontraremos distintos tipos de ejercicios que al presionarles abrirán un enlace en YouTube explicando perfectamente el ejercicio a realizar:



El boton Dietas tiene el mismo funcionamiento que el de Guia Ejerciccios pero con una URL que te envia a una página donde puedes comprar dichos suplementos:




21:24
FitWork
Creatina
Tecnicamente la creatina es un acido organico nitrogenado, es junto con la proteína el suplemento mas utilizado en el mundo. La creatina esta catalogado con eficacia 1A, esta en lo mas alto de la eficacia y seguridad demostrada en multitud de estudios científicos. Sus mayores beneficios se obtienen en deportes de fuerza, ya que aumenta la fuerza y la masa muscular. La creatina es uno de los mejores suplementos para gimnasio y amantes del gym.
¿Como tomar la creatina?
Tomar una dosis de 5 gr 1h antes de hacer ejercicio.
Aconsejablemente tomar un mes si, un mes no.
COMPRAR CREATINA
21:24
www.myprotein.es/nutricio
MYPROTEIN
¡CELEBRAMOS NUESTRO 16º CUMPLEAÑOS!
40% EN TODO, TODO Y TODOOO | CÓDIGO: CUMPLEAÑOS
Página Principal > Creatina Monohidrato
Creatina Monohidrato
Se ha probado científicamente que potencia tu rendimiento
Vegetariano
Vegan
Sin Gluten
300 Reseñas
7,99 €
MYPROTEIN
CREATINE

En el boton Rutinas podemos encontrar distintos tipos de niveles donde puedes realizar una pequeña rutina despende del nivel en el que crear encontrarte:

En el botón de Retos, podemos encontrar una serie de retos para la gente o usuarios más exigente o que simplemente quieren mejorar algún aspecto, sigue siendo un RecyclerView per dentro hay distintas tablas con los días de la semana, series y repeticiones de cada reto:

7. SISTEMA DE GESTIÓN EMPRESARIAL

El sistema de gestión empresarial que nuestra empresa va a utilizar es Odoo.

Odoo es una potente herramienta ERP en la nube que cuenta con aplicaciones de código abierto dirigido a empresas que cubre todas las necesidades de su negocio.

Odoo cuenta con una versión comunitaria de código abierto y una versión empresarial bajo licencia comercial que complementa la edición comunitaria con características y servicios comerciales.

En resumen, es un software empresarial todo en uno que incluye CRM, sitio web y comercio electrónico, facturación, contabilidad, fabricación, gestión de almacenes y proyectos, e inventario entre otros.

Odoo también cuenta con un foro o tienda comunitaria, donde los usuarios crean distintos módulos o aplicaciones que piensan que pueden ser útiles o que les falte a Odoo, por lo que es un Sistema de Gestión Empresarial que siempre está en constante evolución y cambio.

Oficialmente cuenta con más de 46 módulos y más de 20.000 módulos de terceros o de otros usuarios.

Hay que comentar que Odoo es multiplataforma y que anteriormente se le conocía con el nombre de OpenERP.

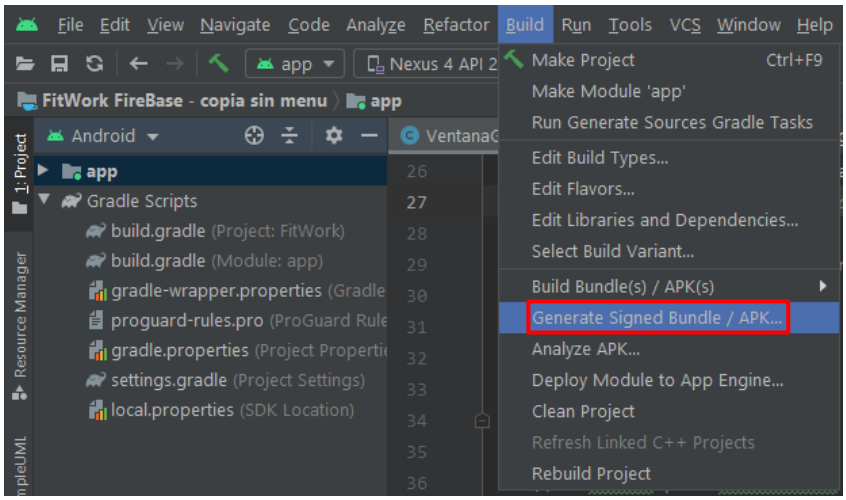
8. DOCUMENTACIÓN

En este apartado voy a explicar cómo generar nuestra firma digital en Android Studio para poder crear nuestra propia APK y así instalar nuestra aplicación en los dispositivos SmartPhones, también mostraré como se administra el foro y a los usuarios de la aplicación mediante FireBase:

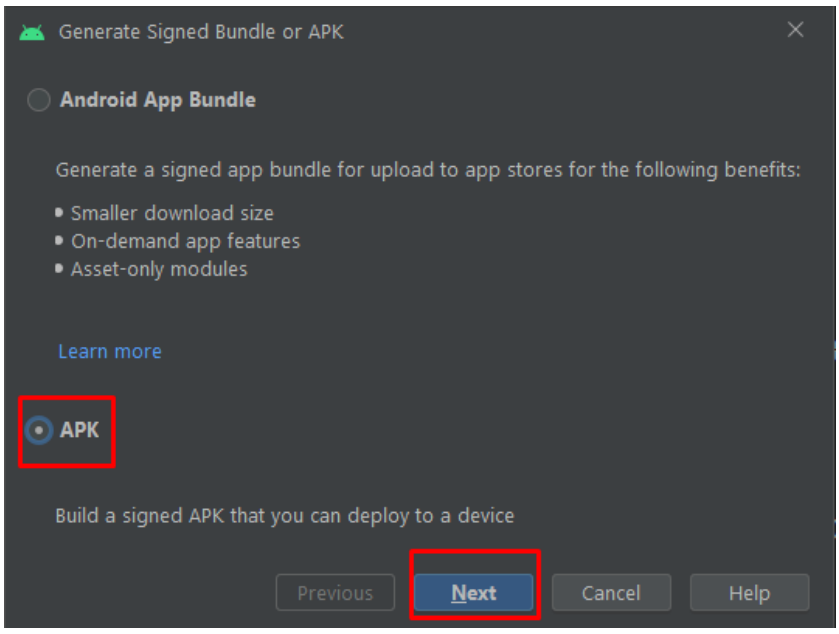
8.1. Manual de instalación

Como ya he explicado antes, vamos a enseñar cómo crear nuestra propia APK para poder instalar la aplicación en los dispositivos móviles:

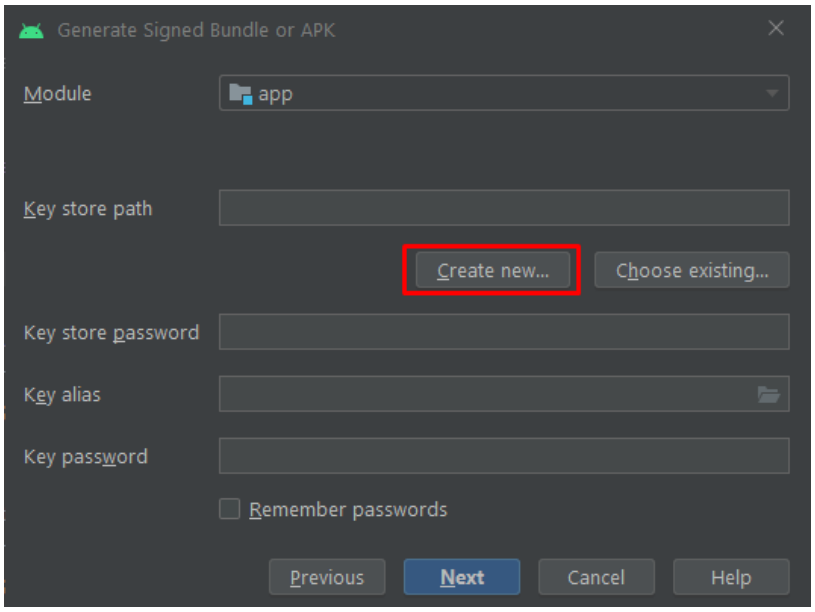
Nos dirigimos a la pestaña Build de Android Studio y seleccionamos Generate Signed Bundle / APK.



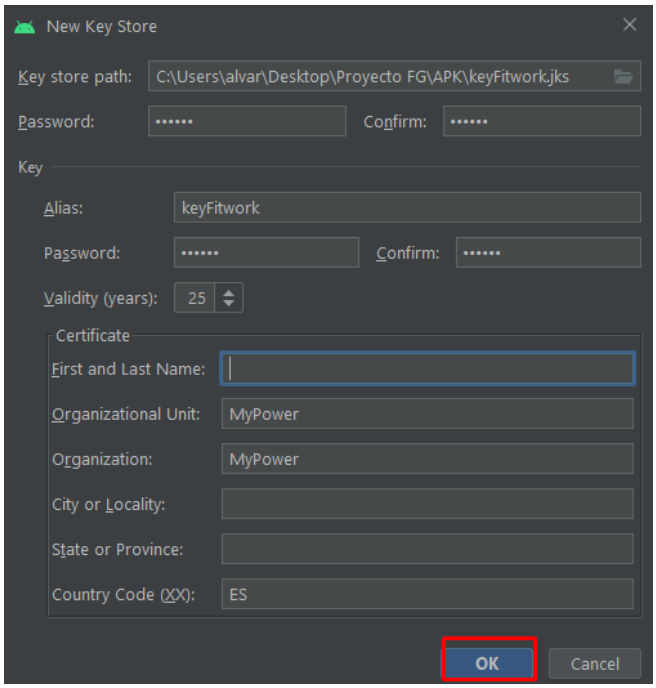
Seleccionamos la opción APK y siguiente.



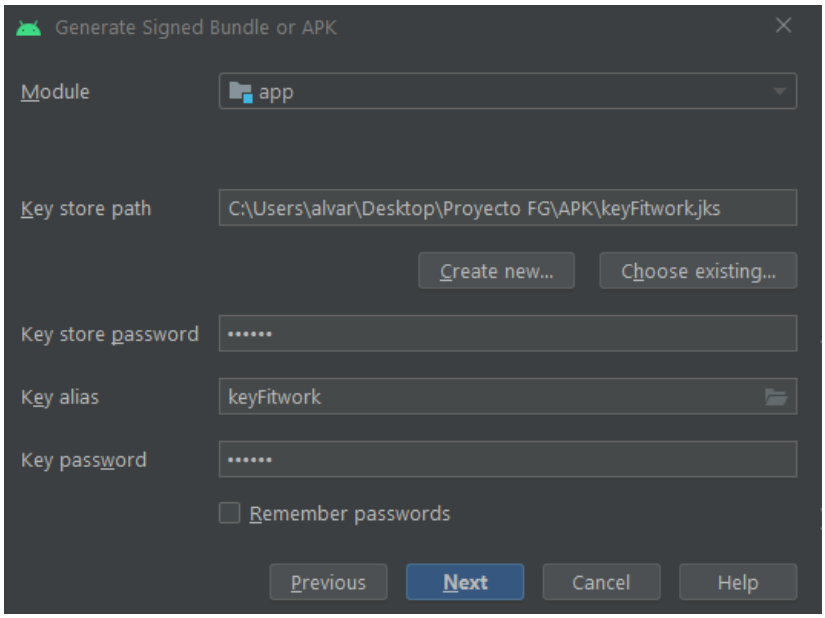
Creamos una nueva ruta para guardar la key.



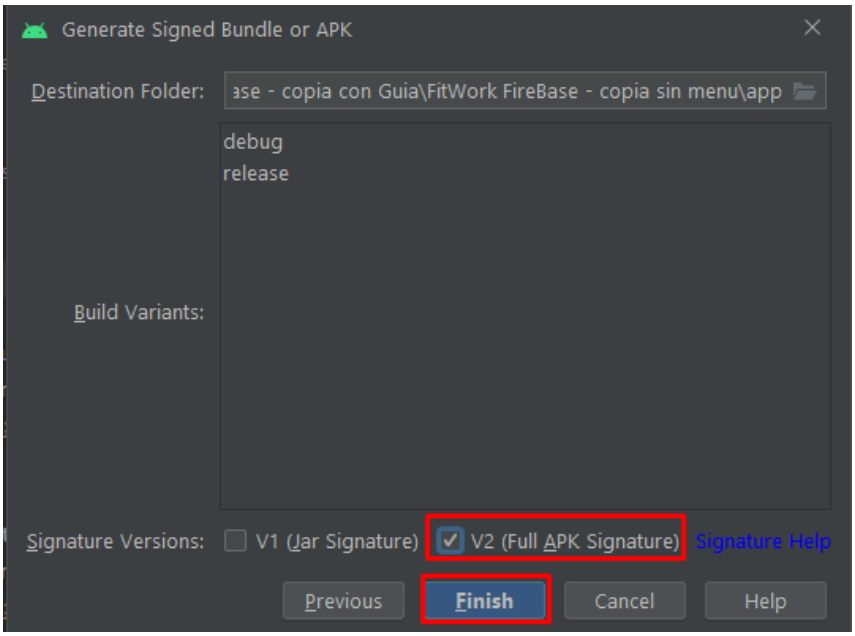
Rellenamos los campos necesarios.



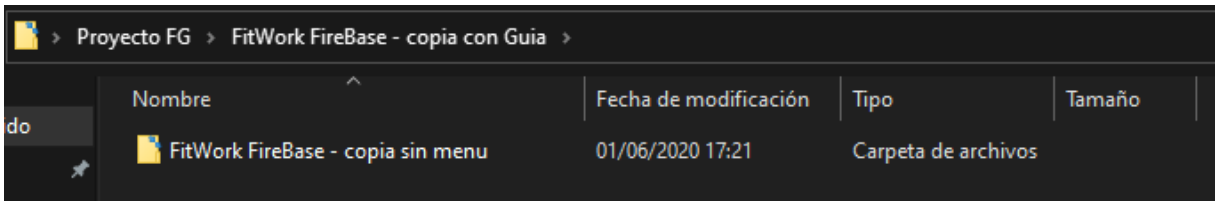
Se nos rellenan los datos de la key creada automáticamente.

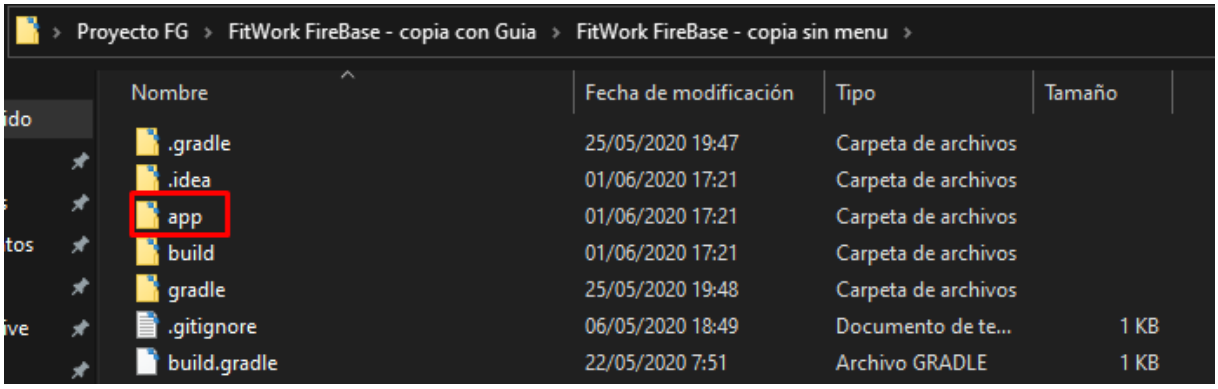


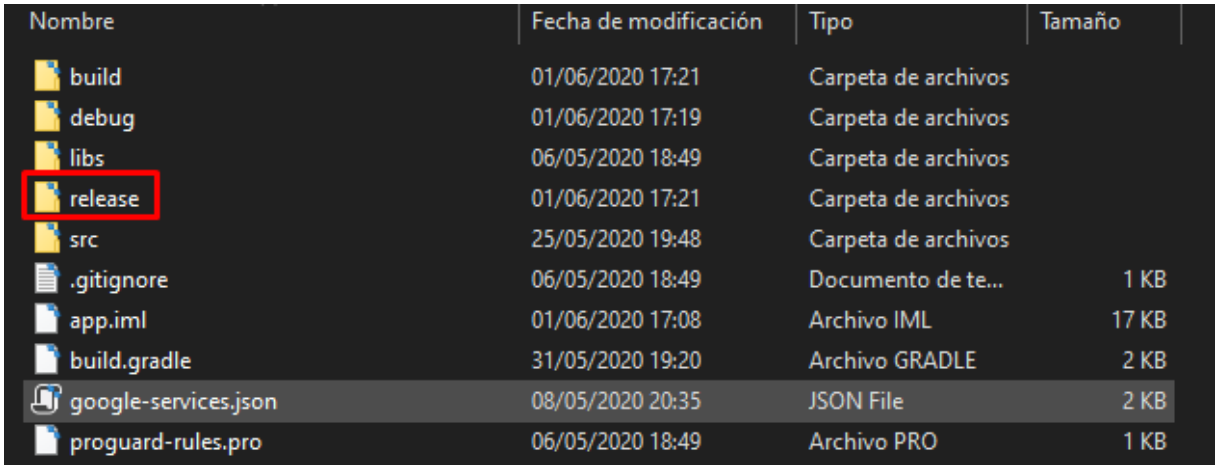
Seleccionamos la versión 2 (v2 full apk signature), para que la firma de la APK sea más segura y finalizamos.

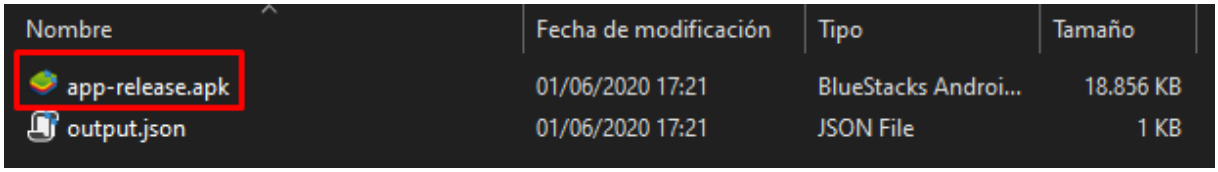


Nos vamos a la carpeta del proyecto, entramos en la carpeta app y seguidamente entramos en la de reléase, dentro tendremos la APK de nuestra aplicación lista.





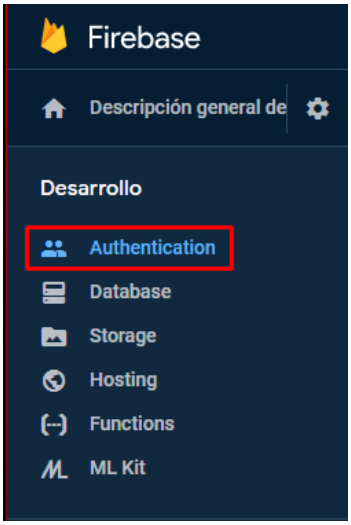




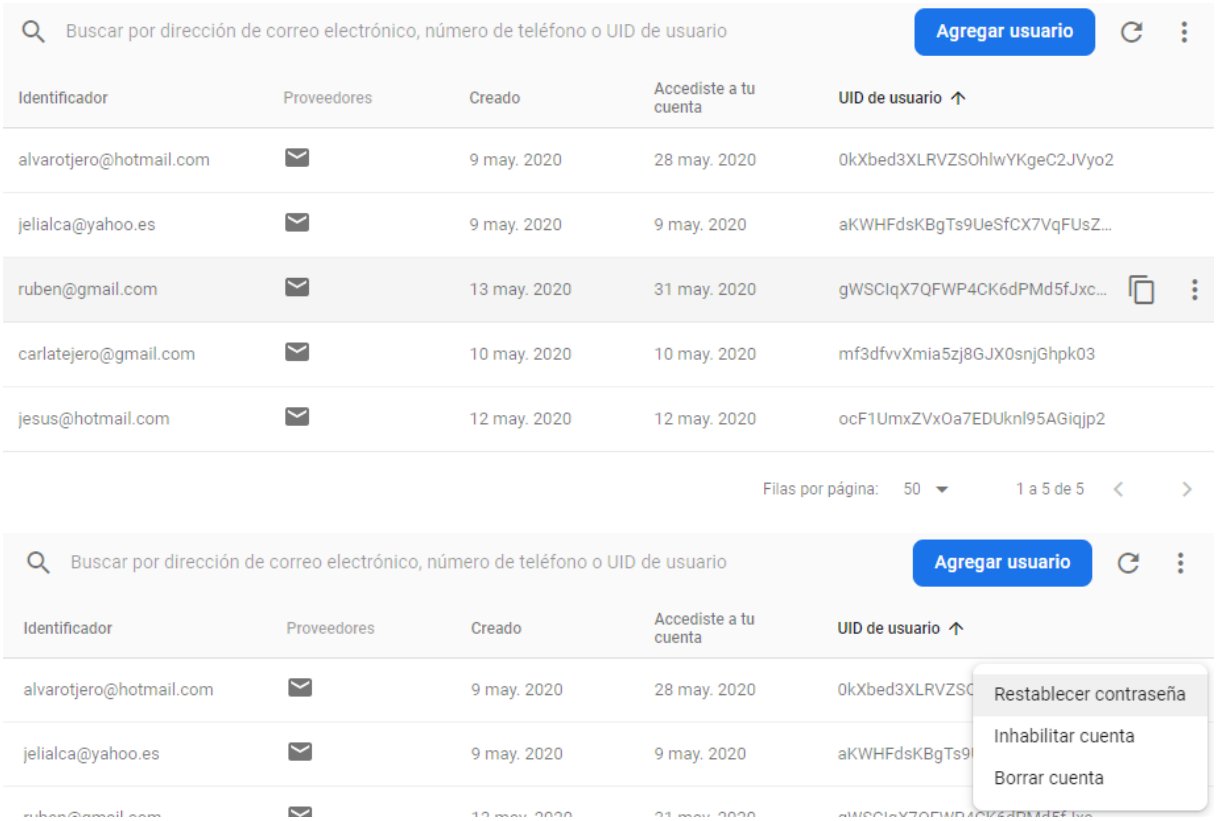
8.2. Administración desde FireBase

Voy a enseñar las ventanas de administración del foro y de usuarios desde FireBase.

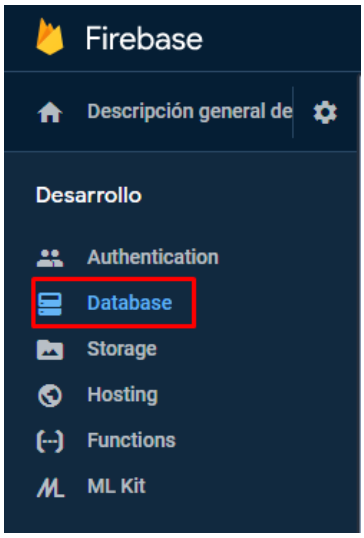
Para administrar los usuarios nos dirigimos a la pestaña de Authentication.



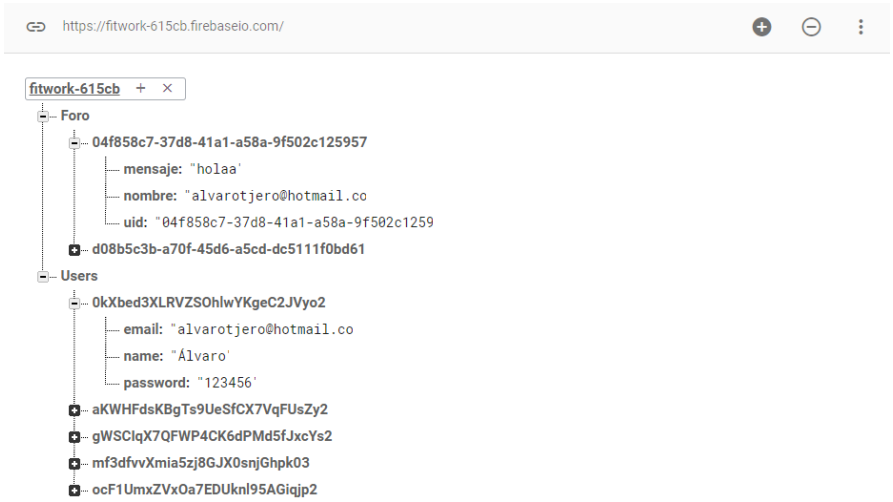
Dentro tenemos una vista de todos los usuarios registrados, podemos agregarlos manualmente, copiar el uid, restablecer la contraseña, inhabilitar la cuenta o borrarla.



Si nos dirigimos a la pestaña de Database podemos monitorizar nuestra base de datos.

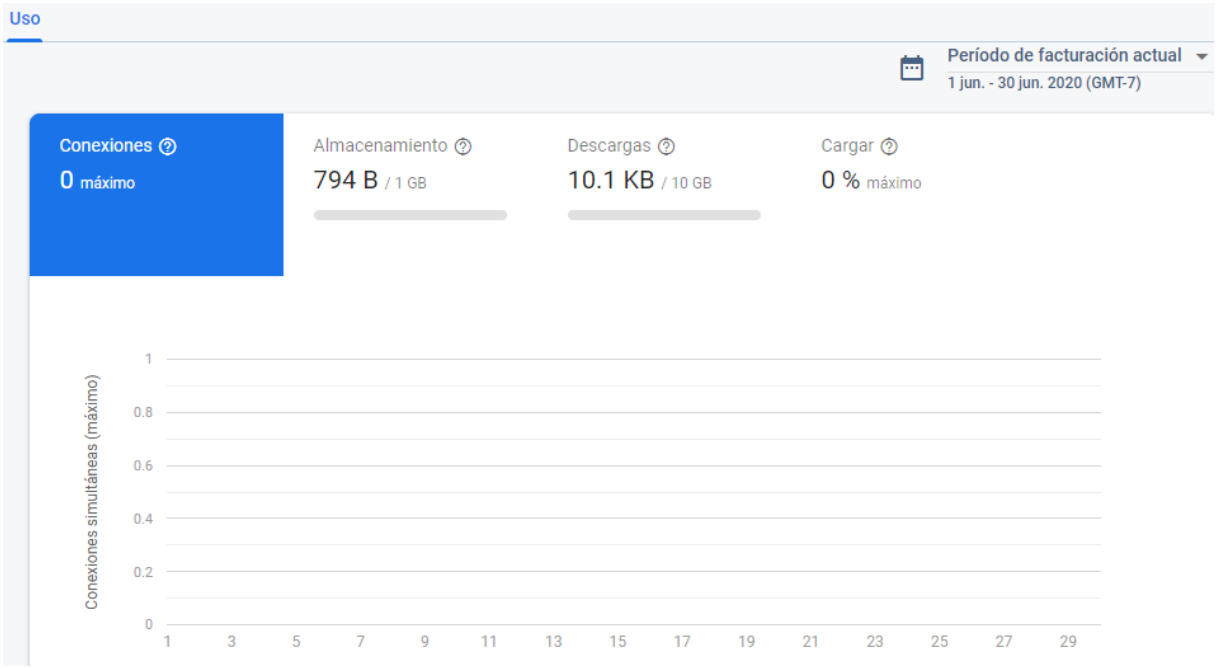


Podemos ver los distintos mensajes del foro con su información asi como la de los usuarios.



En la pestaña de reglas podemos visualizar las reglas de la base de datos, es decir quién puede leer y escribir.

Si entramos en la pestaña de uso, podemos ver las conexiones de los usuarios, el almacenamiento de la base de datos, las descargas y las cargas hechas en la base de datos, así como el uso y la factorización.



9. CONCLUSIONES

En este último apartado aportaré mi sincera opinión sobre el proyecto realizado.

Pienso que se han cumplido los objetivos y requisitos designados para el proyecto, tenía muchas más cosas en mente como una tienda, que en foro apareciera la foto de perfil de cada uno, rutinas personalizadas etc. Por falta de tiempo y de conocimientos no he podido desarrollar todas mis ideas, pero la verdad que al principio del proyecto no sabía ni por dónde empezarlo, gracias a el proyecto pues he descubierto la plataforma de Firebase que es muy útil y puede ahorrar mucho tiempo a la hora de configurar una base de datos para un proyecto, claramente no tenía ni idea ni de cómo utilizarla ni de cómo podía sacarla provecho, por lo que me he tenido que ver y leer un sinfín de videos, tutoriales, foros y documentación para poder desarrollar este proyecto.

Y ¿Por qué escogí Firebase cuando no tenía ni idea de lo que era ni de usarlo?

Ya de por si planear y llevar a cabo tu primer proyecto es una tarea difícil, pero me apetecía aprender todo lo que pudiera durante su desarrollo y si podría haber salido totalmente mal si no me hubiera puesto enserio.

En resumen, la experiencia ha sido dura pero gratificante, es verdad que me hubiera gustado poner muchas más cosas en el proyecto, pero también pienso que para un primer contacto no está mal, estoy orgulloso de que haya podido cumplir con los objetivos iniciales del proyecto y solo es el principio.

10. BIBLIOGRAFIA

Estudio de mercado de sistemas operativos en Smartphones:

https://es.statista.com/estadisticas/600731/cuota-de-mercado-de-sistemas-operativos-para-smartphones-por-pedidos--2020/

Ordenadores PcCom WorkStation:

https://www.pccomponentes.com/pccom-workstation

Red e internet del presupuesto de la empresa:

https://comparaiso.es/ofertas/internet/empresas

Información y precios de Firebase:

https://firebase.google.com/pricing?authuser=0

LucidChart:

https://app.lucidchart.com

Stack Overflow:

https://stackoverflow.com

Sistema de Gestión Empresarial Odoo

https://www.odoo.com/es_ES/

ANEXO DIAGRAMA DE CLASES