



CONFIAX

CICLO: TS Desarrollo de Aplicaciones Multiplataforma

Autor: Ana Malla Espinar

Tutor: Alberto Castedo Espeso

Curso académico 2023-2024

I. RESUMEN

Confiax es una app para móviles que tiene como público objetivo al sector femenino. Permite a la usuaria registrar la fecha de su último periodo y calcular las próximas fechas de su ciclo menstrual en base a la información introducida en la interfaz de usuario y a una serie de cálculos aritméticos. También se ofrece la posibilidad de calcular los días de ovulación y de mayor fertilidad. Es una app muy intuitiva y sencilla que facilita toda la información de un solo vistazo a través de un calendario.

En definitiva, permite a la usuaria registrar, monitorizar y, a partir de los datos introducidos por esta y de los algoritmos programados, estimar las fechas de los futuros ciclos menstruales, de ovulación y de fertilidad.

II. ÍNDICE

I.	<u>RESUMEN</u>	pág. 2
II.	<u>ÍNDICE</u>	pág.3
III.	<u>JUSTIFICACIÓN DEL PROYECTO</u>	pág.4
IV.	<u>INTRODUCCIÓN</u>	pág.6
V.	<u>OBJETIVOS</u>	pág.9
VI.	<u>DESARROLLO</u>	pág.10

VI.I. TECNOLOGÍAS EMPLEADAS

VI.I.I. IMPLEMENTACIÓN DE LA BBDD

VI.I.II. SERVICIO DE AUTENTICACIÓN A TRAVÉS DE FIREBASE

VI.I.III. IMPLEMENTACIÓN DE LA AUTENTICACIÓN CON LA CUENTA DE GOOGLE A TRAVÉS DE FIREBASE EN UNA APLICACIÓN ANDROID

VI.I.IV. IMPLEMENTACIÓN DE LA AUTENTICACIÓN CON LA CUENTA DE FACEBOOK A TRAVÉS DE FIREBASE EN UNA APLICACIÓN ANDROID

VI.I.V. FIRESTORE DATABASE

VI.II. PANTALLAS DE LA APLICACIÓN

VI.III. REQUISITOS

VI.IV. CASOS DE USO

VI.V. DIAGRAMA DE E/R

VI.VI. DIAGRAMA DE CLASES

VI.VII. DIAGRAMA DE SECUENCIA



VII. <u>CONCLUSIONES</u>	<i>pág.51</i>
VIII. <u>CITACIONES Y BIBLIOGRAFÍA</u>	<i>pág.53</i>
IX. <u>ANEXOS</u>	<i>pág.54</i>

III. JUSTIFICACIÓN DEL PROYECTO

Contexto:

En la actualidad, la salud femenina es un aspecto fundamental y cada vez más reconocido en el ámbito de la atención médica y el bienestar personal. El seguimiento preciso y la comprensión de los ciclos menstruales son elementos esenciales para las mujeres en todas las etapas de sus vidas, ya sea para la planificación familiar, la salud reproductiva o simplemente para mantener un control integral de su bienestar.

Problemática:

A pesar de la importancia de este seguimiento, muchas mujeres enfrentan dificultades para llevar un registro exacto y detallado de sus ciclos menstruales. La falta de herramientas accesibles, intuitivas y personalizadas contribuye a la falta de conciencia y comprensión sobre la salud menstrual.

Objetivos:

El objetivo principal de este proyecto es desarrollar una aplicación móvil centrada en el usuario que facilite la monitorización, notificación y registro de ciclos menstruales. La aplicación busca proporcionar a las usuarias una herramienta integral y fácil de usar que les permita:

- **Registro Preciso:** llevar un registro detallado de los eventos menstruales, incluyendo fechas de inicio y finalización, síntomas y otros aspectos relevantes.
- **Monitorización Personalizada:** proporcionar funciones de monitorización que permitan a las usuarias visualizar patrones, identificar tendencias y comprender mejor su salud reproductiva.
- **Notificaciones Informadas:** enviar notificaciones oportunas sobre eventos críticos como el inicio de la menstruación, ovulación y días fértiles, contribuyendo así a una mayor conciencia y planificación.
- **Facilitar la Comunicación con Profesionales de la Salud:** ofrecer la capacidad de exportar datos y compartir información relevante con profesionales de la salud, fomentando una comunicación más efectiva durante consultas médicas.

- **Beneficios Esperados:** el desarrollo de esta aplicación tiene el potencial de impactar positivamente la salud y el bienestar de las usuarias al proporcionarles una herramienta práctica y personalizada para el seguimiento de sus ciclos menstruales. Se espera que la aplicación contribuya a una mayor conciencia de la salud femenina, apoye la toma de decisiones informadas y fortalezca la relación entre las usuarias y sus profesionales de la salud.
- **Consideraciones Éticas y de Privacidad:** hay una intención de compromiso para implementar rigurosas medidas de seguridad y privacidad para garantizar la protección de los datos sensibles de las usuarias, cumpliendo con las normativas y estándares éticos más altos.

En resumen, el desarrollo de esta aplicación responde a la necesidad crítica de ofrecer a las mujeres una herramienta moderna y efectiva para el seguimiento de sus ciclos menstruales, contribuyendo así a la promoción de la salud femenina.

IV. INTRODUCCIÓN

La sociedad actual, marcada por un ritmo trepidante, a menudo impide que las personas dediquen el tiempo necesario para atender su salud. En el caso particular de las mujeres, este desafío se agrava al tener que gestionar la complejidad de su salud ginecológica y sexual. Las demandas laborales, los desplazamientos diarios entre la casa y la oficina, las responsabilidades familiares, entre otros, relegan la atención al funcionamiento del órgano reproductor femenino. Este seguimiento es crucial desde la perspectiva ginecológica para detectar posibles alteraciones, como cambios en la duración de los días de sangrado o en el intervalo entre menstruaciones, que podrían ser indicativos de patologías.

En este contexto, el uso de una aplicación que ofrezca servicios de registro y consulta mensual de menstruaciones, así como el cálculo de periodos de ovulación y fertilidad, simplifica enormemente el seguimiento de la salud sexual de la mujer, brindando comodidad e inmediatez. Además, la aplicación presenta ventajas económicas y ambientales en comparación con los métodos tradicionales, como el uso de papel, bolígrafos o agendas. En el método convencional, localizar datos de menstruaciones pasadas puede resultar complicado, y los registros físicos ocupan espacio y tienden a ser desechados, lo que implica un desperdicio innecesario. Se trata de un sistema poco práctico, eficiente y no amigable con el medio ambiente.

En la actualidad, el mercado ofrece una amplia variedad de aplicaciones móviles dedicadas a este propósito, brindando a los usuarios numerosas opciones según sus necesidades y preferencias. La información necesaria para el desarrollo del proyecto, desde requisitos hasta análisis, diseño e implementación del código de la aplicación, se encuentra fácilmente disponible en repositorios, sitios académicos, páginas especializadas en salud femenina, y documentación de aplicaciones populares de seguimiento del ciclo menstrual. La accesibilidad y diversidad de estas aplicaciones en la tienda virtual *Play Store* facilitan el acceso a herramientas efectivas para el monitoreo y la gestión de la salud menstrual.



WomanLog



Clue



Flo

Las aplicaciones para el seguimiento del ciclo menstrual han experimentado un crecimiento significativo en el mercado de la salud femenina. Ejemplos notables como **WomanLog**, **Clue** y **Flo**

han ganado popularidad y han establecido estándares en la industria. Aquí hay algunas características comunes y aspectos relevantes sobre estas aplicaciones:

- Registro Menstrual:

Estas aplicaciones permiten a las usuarias realizar un seguimiento detallado de su ciclo menstrual, incluyendo fechas de inicio y finalización, síntomas, cambios de humor y otros aspectos relacionados.

- Predicciones de Ovulación y Fertilidad:

Proporcionan estimaciones de los días de ovulación y fertilidad basadas en algoritmos que consideran la duración del ciclo menstrual y otros factores ingresados por la usuaria.

- Notificaciones y Recordatorios:

Envían alertas y recordatorios sobre la llegada de la menstruación, días fértiles, ovulación, y otros eventos relevantes del ciclo menstrual.

- Personalización y Análisis:


Ofrecen herramientas para personalizar la experiencia del usuario, adaptando la aplicación a las necesidades y preferencias individuales. Además, proporcionan análisis visuales y estadísticas sobre los patrones menstruales a lo largo del tiempo.

- Integración con Dispositivos y Plataformas:

Algunas aplicaciones ofrecen integración con dispositivos de seguimiento de salud y sincronización con otras plataformas para una experiencia más completa.

- Privacidad y Seguridad:

Estas suelen enfocarse en la privacidad y la seguridad de los datos, permitiendo a las usuarias controlar quién accede a su información.

En lo que respecta a **Confiax**  la implementación rigurosa de los algoritmos encargados de calcular los periodos menstruales, de ovulación y de fertilidad se ha llevado a cabo mediante consultas de diversas fuentes en línea especializadas en ginecología. Prácticamente todas estas fuentes convergen casi al 100% en la aplicación de cálculos matemáticos específicos. Esta consistencia ha simplificado considerablemente la tarea de programar estas operaciones aritméticas.

Sin embargo, es importante destacar que, dado que no se busca aplicar un enfoque de talla única a todas las mujeres, en la aplicación **Confiax** se ha incorporado la opción para que la usuaria final introduzca sus propios datos menstruales. Esto permite obtener resultados personalizados, fiables y precisos que se adaptan a sus patrones y características individuales. Al ofrecer esta personalización, se asegura así un servicio adaptado a sus necesidades específicas.

En conclusión, con el desarrollo de **Confiax** se busca reunir esas características que la hagan triunfar en el universo de la apps de seguimiento del ciclo menstrual: una interfaz intuitiva; un registro rápido; predicciones precisas; funciones de recordatorio y notificación que alerten a la usuaria; la capacidad de personalizar la aplicación según las preferencias individuales de la usuaria; un historial completo y funciones de análisis que permitan a la usuaria revisar patrones a lo largo del tiempo; un diseño atractivo y amigable; garantías de privacidad y seguridad de los datos y, finalmente, disponibilidad en múltiples plataformas, como iOS y Android

V. OBJETIVOS

El proyecto tiene por objetivo crear una aplicación Android para:

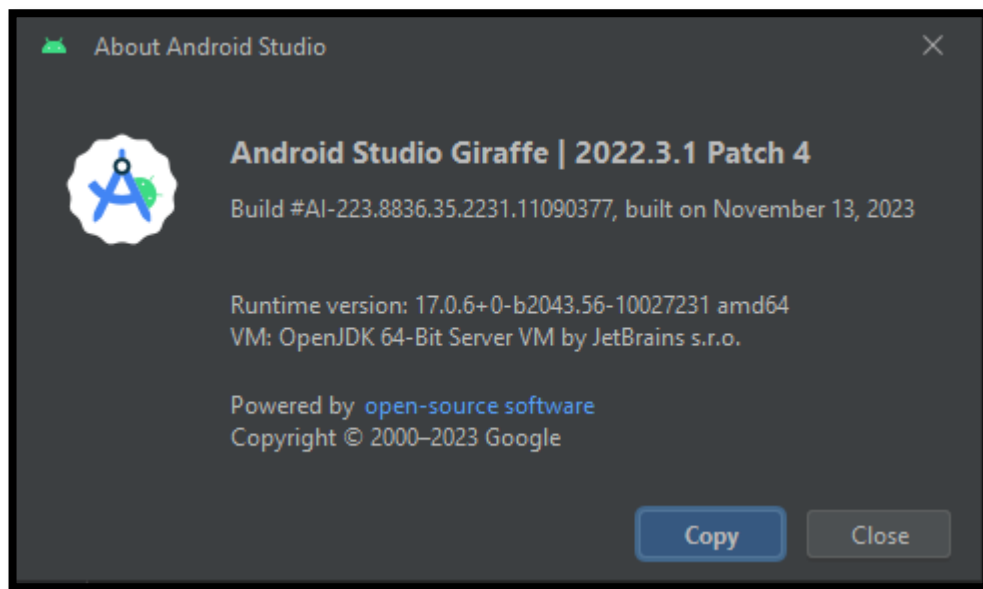
- ✓ Registrar la fecha de la última menstruación de la usuaria.
- ✓ Registrar la duración del periodo.
- ✓ Registrar el espacio de tiempo entre menstruaciones.
- ✓ Seleccionar la opción de notificación de la próxima menstruación.
- ✓ Seleccionar la opción de notificación de los días fértiles.
- ✓ Seleccionar la opción de notificación de días de ovulación.
- ✓ Calcular la próxima menstruación.
- ✓ Calcular la próxima ovulación.
- ✓ Calcular los días de mayor fertilidad.
- ✓ Visualizar en un calendario interactivo el histórico de fechas de menstruaciones, ovulaciones y días fértiles y también las estimaciones de las próximas.

Con el fin de cumplir dichos objetivos, se requiere conocimientos en BBDD, programación en Java y en el desarrollo de interfaces graficas en dispositivos móviles.

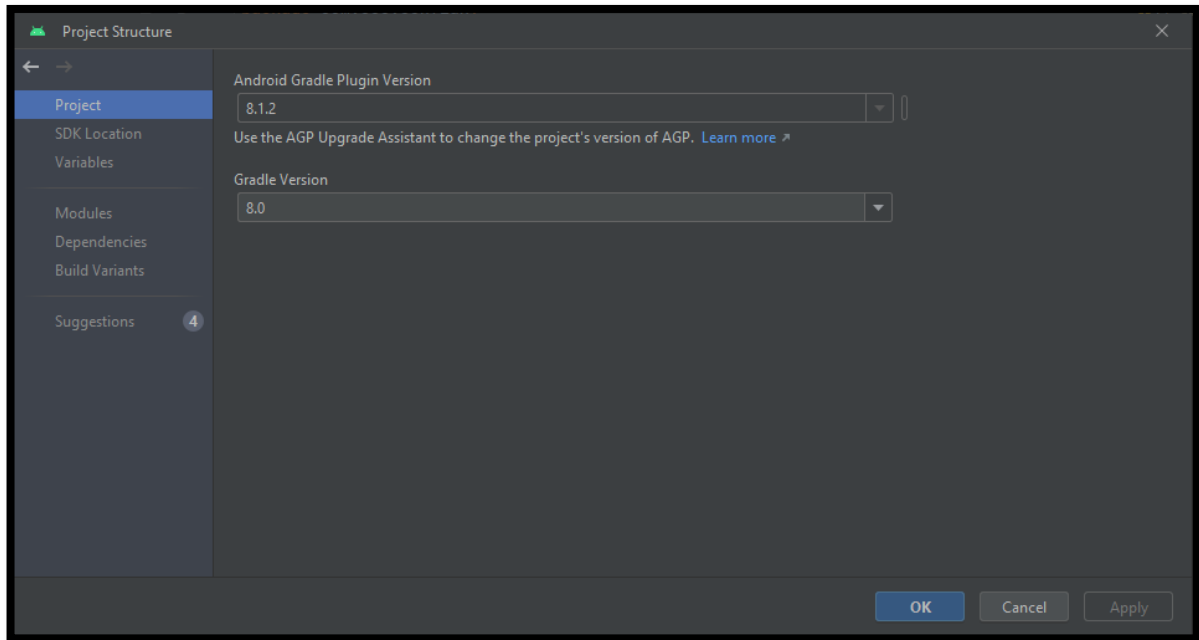
VI. DESARROLLO

VI.I. TECNOLOGÍAS EMPLEADAS

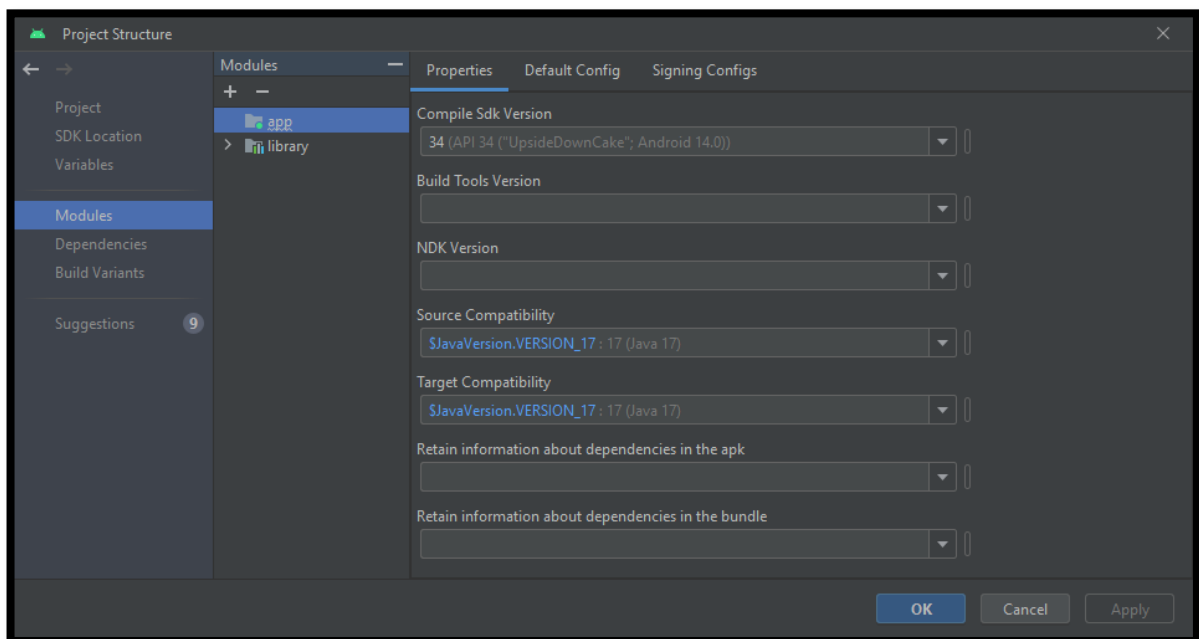
Confiar se desarrolla para el S.O. Android. Este fue desarrollado por Google y lanzado en 2008, basado en Kernel de Linux y otros softwares de código abierto. Fue diseñado para dispositivos móviles con pantalla táctil, como smartphones, tablets, smartwatches, automóviles y televisores.



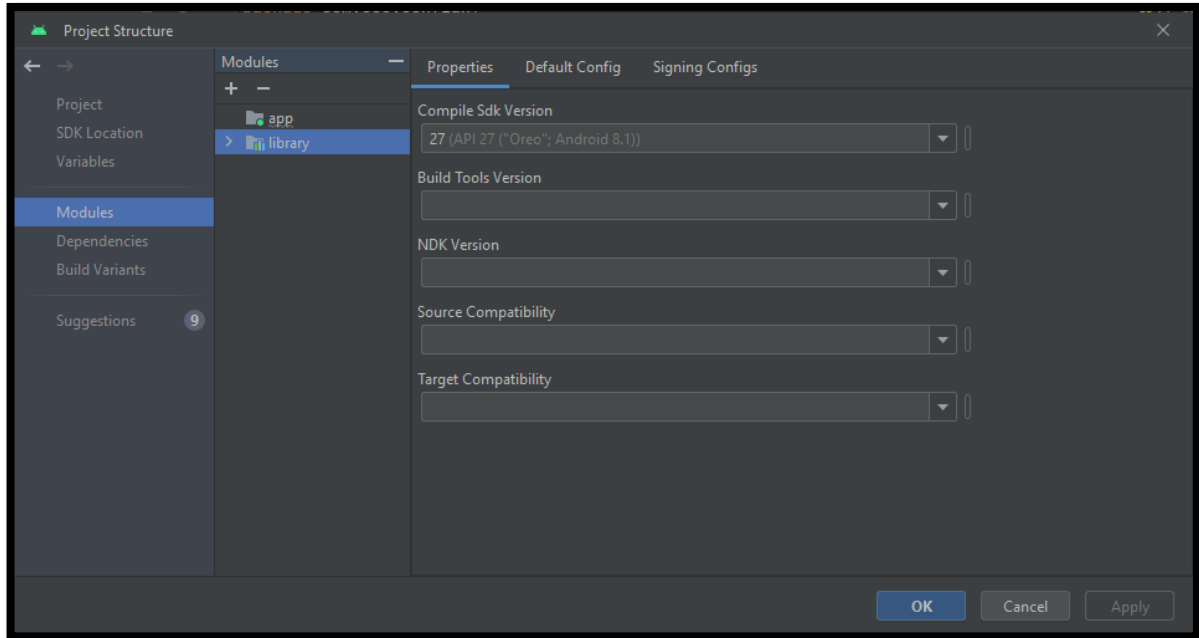
- ✓ Como lenguaje de programación se ha empleado Java versión 17.
- ✓ Como IDE se ha usado Android Studio Giraffe | 2022.3.1 Patch 4
- ✓ VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o.
- ✓ Windows 10 10.0
- ✓ Memory: 2048M
- ✓ Cores: 4



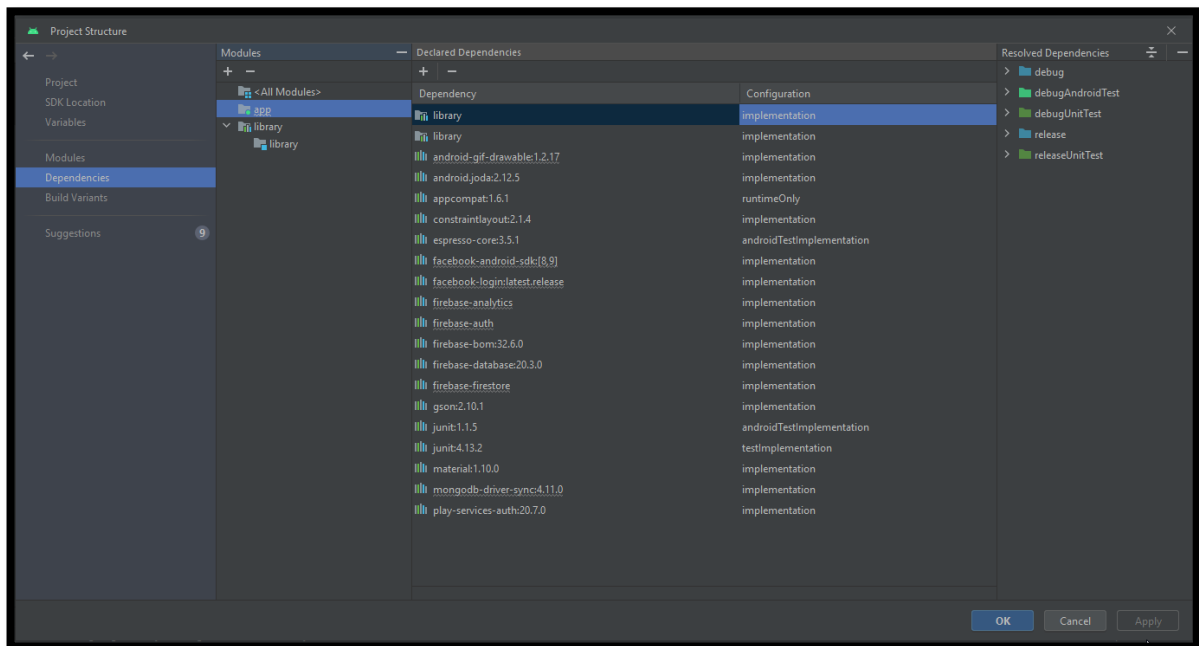
Se ha usado Gradle version 8 (<https://docs.gradle.org/8.0/release-notes.html>)



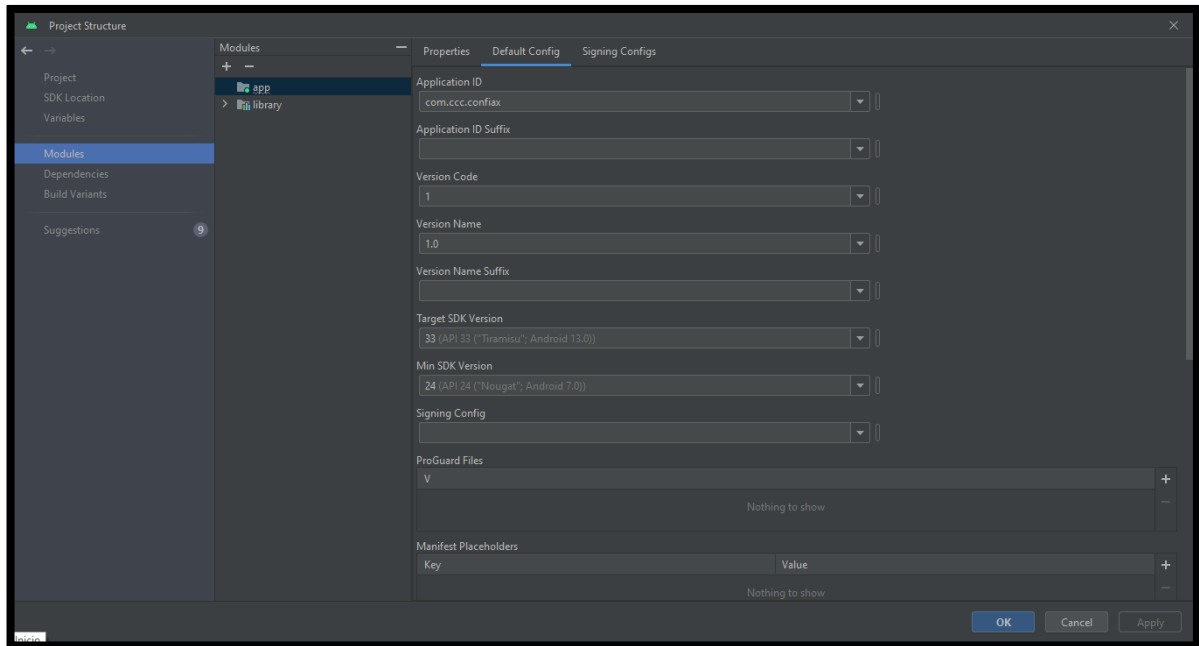
La versión del SDK de Android que se ha usado para compilar la app es la 34



La librería del “CustomCalendarView” usa la SDK versión 27 (<https://github.com/StackTipsLab/custom-calendar-view>)



Lista de dependencias de las librerías del proyecto



Configuración general de la app

VI.I.I. IMPLEMENTACIÓN DE LA BBDD

En principio se pensó en **MongoDB**, que es un sistema de base de datos NoSQL (*Not only SQL*), orientado a documentos y de código abierto.



La ventaja que tienen las NoSQL frente a las clásicas SGBDR (Sistema de Gestión de bases de datos relacionales) es:

- ✓ Están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas.
- ✓ Son ampliamente reconocidas porque son fáciles de desarrollar, tanto en funcionalidad como en rendimiento a escala.
- ✓ Usan una variedad de modelos de datos, que incluyen documentos, gráficos, clave-valor, en-memoria y búsqueda.
- ✓ La información se organiza normalmente mediante documentos y es muy útil cuando no tenemos un esquema exacto de lo que se va a almacenar.

Se instaló el producto (MongoDB), la herramienta de la Shell (**MongoShell**) y **MongoDB Compass**, que es el software oficial desarrollado por Mongo para la visualización de bases de datos. También creé un clúster de bases de datos en **MongoDB Atlas**, que es un conjunto integrado de BBDD de cloud y servicios de datos, para conectarme a él desde MongoDB Compass.

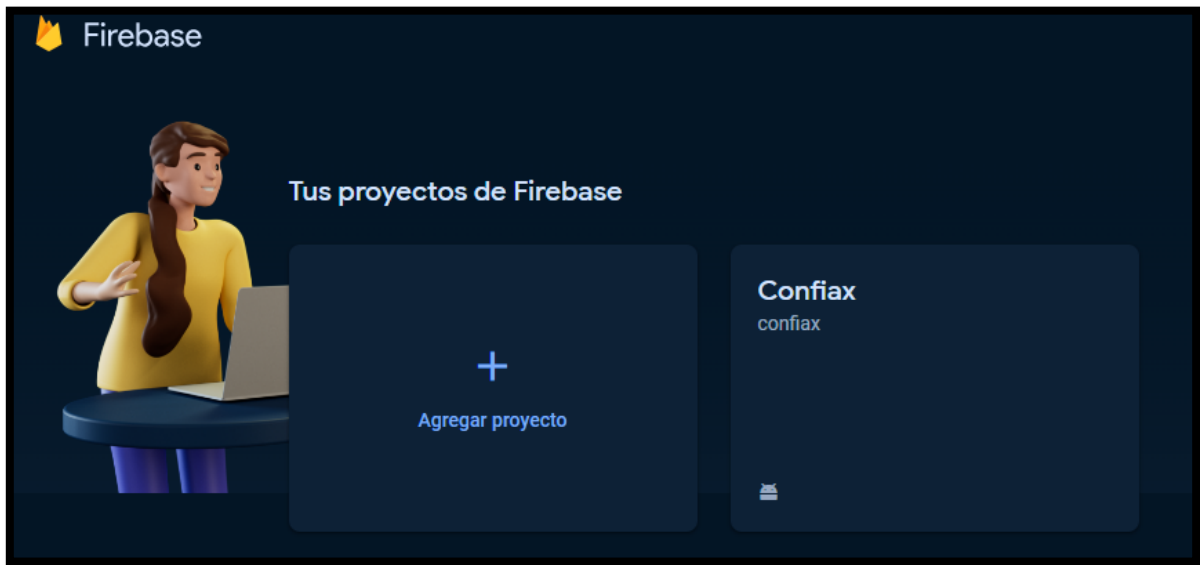
```
android.support.test.espresso:espresso-core:3.3.1")
implementation("pl.droidsonroids.gif:android-gif-drawable:1.2.17")
implementation("org.mongodb:mongodb-driver-sync:4.11.0")
implementation(platform("com.google.firebase:firebase-bom:32.6.0"))
implementation("com.google.firebase:firebase-auth:21.0.0")
```

Sin embargo, una vez que estuvo todo instalado y configurado, hubo una serie de problemas con la gestión de la conexión a la hora de recuperar la información de la colección creada, 'User'.

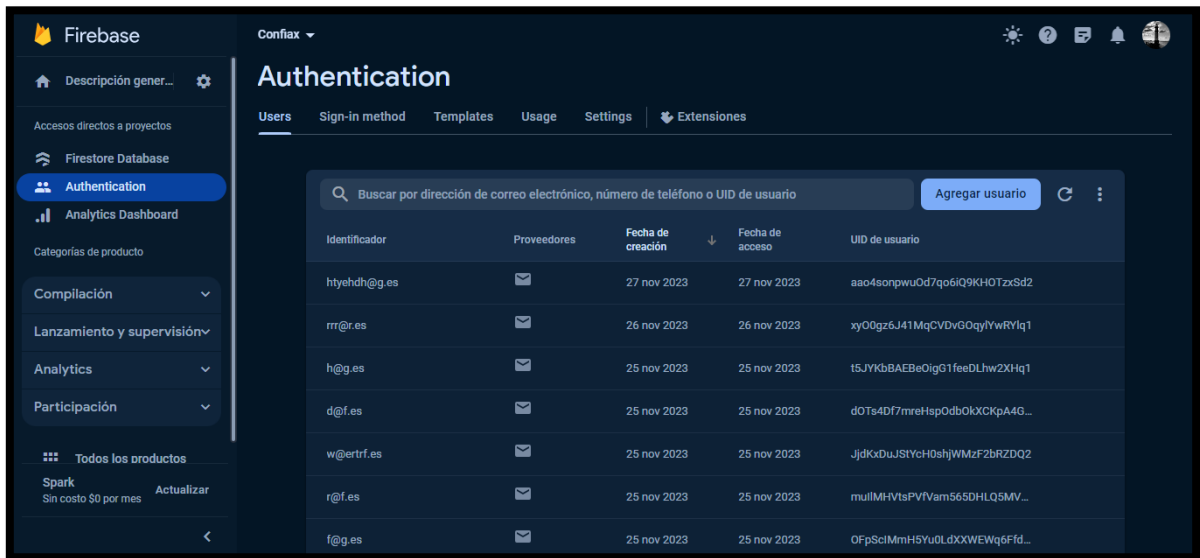
Finalmente se optó por la plataforma en la nube de **Firestore de Google**.



- ✓ En Firebase se dio de alta la app para Android **Confiax**.



Inicio de la plataforma *Firebase*



Consola de administración de *Firebase*

- ✓ Una vez agregada y configurada la aplicación en la plataforma, se descargó el fichero de configuración de Firebase: *google-services.json*. Este archivo contiene la información de configuración necesaria para que la aplicación se comuniquen con Firebase.
- ✓ Se agregó el archivo de configuración al proyecto **Confiax** (a nivel del directorio 'app').

- ✓ Se agregó las dependencias de Firebase: en el archivo *build.gradle* (módulo de la aplicación), se declararon las dependencias necesarias. Estas dependencias están relacionadas con los servicios de Firebase que se van a usar.

```
implementation(platform("com.google.firebase:firebase-bom:32.6.0"))
```

En la misma web de Firebase proponen que se declare una dependencia a "*Bill of Materials*" (BOM), de Firebase, que agrupa las versiones de las bibliotecas de esta.

Simplifica la gestión de versiones al centralizar las versiones de las dependencias en un solo lugar. De esta manera, no se necesita especificar explícitamente la versión de cada dependencia de Firebase en el archivo *build.gradle*, ya que el BOM se encarga de eso.

Esto hace que la gestión de versiones sea más sencilla y menos propensa a errores. Si hay una nueva versión de Firebase, sólo se necesitará actualizar la versión del BOM en el archivo *build.gradle*.

- ✓ Se descargó la versión más reciente de las dependencias en la página de GitHub de Firebase.
- ✓ Se sincronizó y compiló el proyecto con Gradle.
- ✓ En ese punto ya estaba agregada la app en la plataforma de Firebase y Firebase en la app.
- ✓ En principio, no se tuvo que inicializar Firebase en la app con el siguiente método:

```
FirebaseApp.initializeApp(this);
```

Se desconoce el impacto de no usar esta sentencia a la hora de actualizar algunas librerías importantes como las de Android o Gradle.

VI.I.II. SERVICIO DE AUTENTICACIÓN A TRAVÉS DE FIREBASE

La autenticación a través de Firebase es un proceso bastante sencillo:

- ✓ Se agregó la dependencia de Firebase Authentication

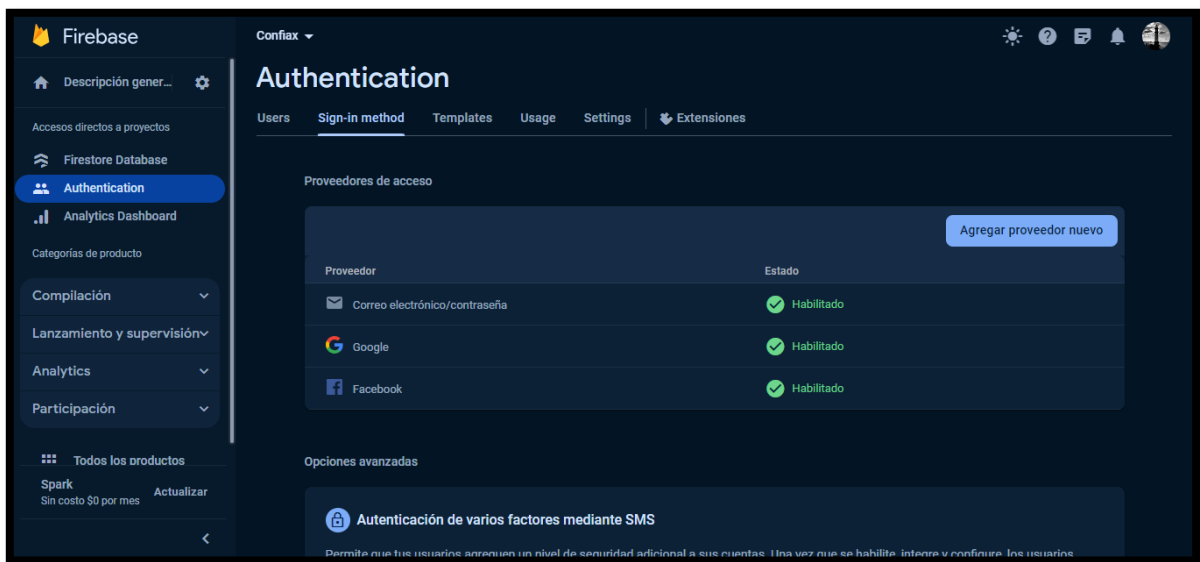
En el archivo build.gradle (módulo de la aplicación), se declaró la dependencia para Firebase Authentication:

```
implementation("com.google.firebase:firebase-auth")
```

- ✓ Se configuraron los proveedores de autenticación

En la Consola de Firebase → sección de Authentication.

En la pestaña "*Método de acceso*", se configuraron dos proveedores de autenticación: **Google** y **Facebook**.



Menú de Autenticación en la consola de *Firebase*

```
1 usage
public void login (String email, String pwd) {
    firebaseAuth.signInWithEmailAndPassword(email, pwd).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if (task.isSuccessful()) {
                //finish();
                // Sign in success, update UI with the signed-in user's information
                Log.d(TAG, "signInWithEmailAndPassword:success");
                FirebaseUser user = firebaseAuth.getCurrentUser();
                updateUI(user);
                startActivity(new Intent( packageContext: LoginActivity.this, HomeActivity.class));
                Toast.makeText( context: LoginActivity.this, text: "¡Bienvenido a Confiax!", Toast.LENGTH_LONG).show();
            } else {
                Toast.makeText( context: LoginActivity.this, text: "Error al iniciar sesión.", Toast.LENGTH_LONG).show();
                /*TODO: pendiente gestionar los errores en el login*/
            }
        }
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            Toast.makeText( context: LoginActivity.this, text: "Error al iniciar sesión", Toast.LENGTH_LONG).show();
            // Si/No
            signup ();
            /*TODO: pendiente gestionar los errores en el login*/
        }
    });
}
```

Implementación de la autenticación en la aplicación: método *login* de la clase *LoginActivity.java*

✓ Manejo de sesiones y cierre de sesión:

- Para cerrar sesión se hace mediante la llamada a `signOut()` en la instancia de `FirebaseAuth`:

```
1 usage
public void logOut(View v) {
    firebaseAuth.signOut();
    Auth.GoogleSignInApi.signOut(googleApiClient).setResultCallback(status -> {
        if (status.isSuccess()) {
            goLogin();
        } else {
            Toast.makeText(y.getContext(), text: "No se pudo cerrar sesión. Vuelva a intentarlo.", Toast.LENGTH_LONG)
        }
    });
}
```

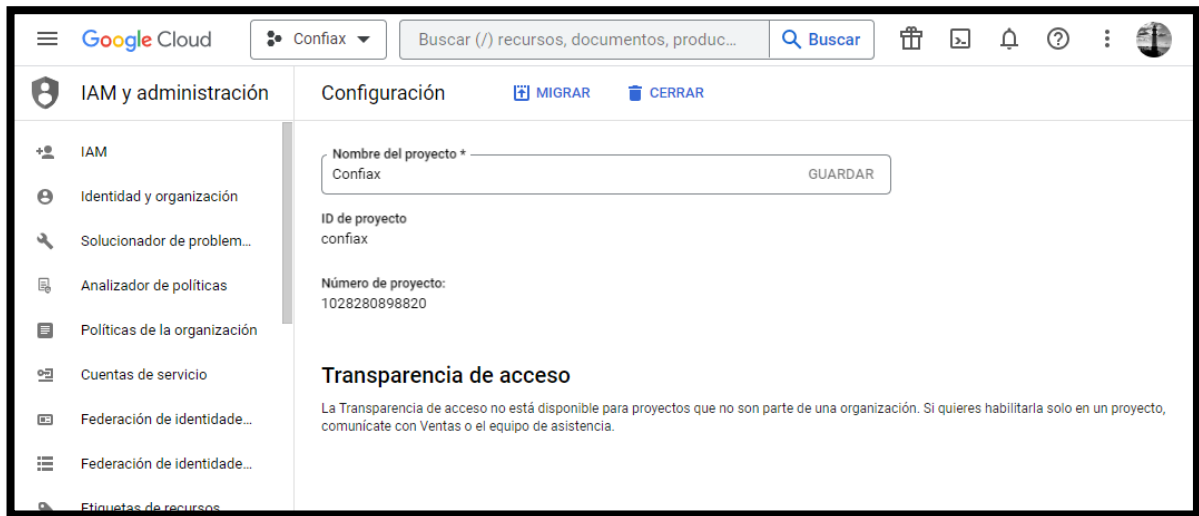
MenstrualCalendarActivity.java

- Para verificar si un usuario ya está autenticado, se usa la función `getCurrentUser()`. Al implementarse el método `signInWithEmailAndPassword()`, el objeto `FirebaseUser` contiene información del UID (identificador único), del email y de la password del usuario.

```
1 usage
public void login (String email, String pwd) {
    firebaseAuth.signInWithEmailAndPassword(email, pwd).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if (task.isSuccessful()) {
                //finish();
                // Sign in success, update UI with the signed-in user's information
                Log.d(TAG, "signInWithEmailAndPassword:success");
                FirebaseUser user = firebaseAuth.getCurrentUser();
                updateUI(user);
                startActivity(new Intent( packageContext LoginActivity.this, HomeActivity.class));
                Toast.makeText( context LoginActivity.this, "¡Bienvenido a Confiax!", Toast.LENGTH_LONG).show();
            } else {
                Toast.makeText( context LoginActivity.this, "Error al iniciar sesión.", Toast.LENGTH_LONG).show();
                /*TODO: pendiente gestionar los errores en el login*/
            }
        }
    })
}
```

LoginActivity.java

VI.I.III. IMPLEMENTACIÓN DE LA AUTENTICACIÓN CON LA CUENTA DE GOOGLE A TRAVÉS DE FIREBASE EN UNA APLICACIÓN ANDROID



- ✓ Configuración de la Consola de Desarrolladores de Google:
 - Se dio de alta la app de Confiar.
 - Se habilitó la API de "Inicio de sesión con Google".
- ✓ Configuración de Credenciales:
 - En la Consola de Desarrolladores de Google, se crearon credenciales para la aplicación Android.
 - Se seleccionó la "ID del cliente de OAuth" y se eligió "Aplicación de Android".
 - Se ingresó el nombre del paquete de aplicación (applicationId = com.ccc.confiar en el archivo build.gradle).

```
defaultConfig { this: ApplicationDefaultConfig
    applicationId = "com.ccc.confiax"
    minSdk = 24
    targetSdk = 33
    versionCode = 1
    versionName = "1.0"
    buildToolsVersion = "34"
    testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
}
```

- Se registró la huella digital de certificación de la clave de firma de la aplicación. Esta se obtuvo ejecutando la siguiente tarea en el terminal:

```
Starting Gradle Daemon...
Gradle Daemon started in 2 s 950 ms

> Task :app:signingReport
Variant: debug
Config: debug
Store: C:\Users\Ana\.android\debug.keystore
Alias: AndroidDebugKey
MD5: [REDACTED]
SHA1: [REDACTED]
SHA-256: [REDACTED]
Valid until: jueves, 30 de octubre de 2053
```

- Se copió el ID del cliente generado.

Additional information

ID de cliente	1028280898820- ut11pdloa99bmbdbmtoqo4lqkmf6cj e.apps.googleusercontent.com
Fecha de creación	20 de noviembre de 2023, 13:27:42 GMT+1

✓ Configuración en Firebase:

- Se volvió a la Consola de Firebase y se seleccionó el proyecto.
- En la sección de autenticación, en la pestaña "Método de acceso", se habilitó

"Google" como método de acceso.

- Se 'pegó' el ID del cliente de OAuth que se copió anteriormente.

✓ Declaración de dependencia de Firebase Auth:

- Se añadió la dependencia de Firebase Authentication en el archivo build.gradle:

```
implementation("com.google.firebase:firebase-auth")
```

✓ Implementación del Inicio de Sesión con Google en tu App:

- Se implementó la lógica de inicio de sesión con Google en la clase **LoginActivity.java**.

✓ Manejo del Resultado del Inicio de Sesión:

- Se implementó el método **onActivityResult(..)** para manejar el resultado del inicio de sesión:

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == SING_IN_CODE) {
        GoogleSignInResult googleSignInResult = Auth.GoogleSignInApi.getSignInResultFromIntent(data);
        handleSignInResult(googleSignInResult);
    }
    callbackManager.onActivityResult(requestCode, resultCode, data);
}
```

✓ Autenticación en Firebase:

- Se implementó el método **firebaseAuthWithGoogle** para autenticar con Firebase:

```
1 usage
private void firebaseAuthWithGoogle(GoogleSignInAccount signInAccount) {

    progressBar.setVisibility(View.VISIBLE);
    btloginGoogle.setVisibility(View.GONE);

    AuthCredential credential = GoogleAuthProvider.getCredential(signInAccount.getIdToken(), null);
    firebaseAuth.signInWithCredential(credential).addOnCompleteListener( activity, this, new OnCompleteListener<AuthResult>() {

        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {

            progressBar.setVisibility(View.GONE);
            btloginGoogle.setVisibility(View.VISIBLE);

            if (!task.isSuccessful()) {
                Toast.makeText(getApplicationContext(), text: "Error en el inicio de sesión", Toast.LENGTH_SHORT).show();
            }
        }
    });
}
```

✓ Manejo del Cierre de Sesión:

- Para el cierre de sesión, se utilizó **GoogleSignInClient** para desconectar al usuario de Google y luego cerrar sesión en Firebase:

```
1 usage
public void logout(View v) {
    firebaseAuth.signOut();
    Auth.GoogleSignInApi.signOut(googleApiClient).setResultCallback(status -> {
        if (status.isSuccess()) {
            goLogin();
        } else {
            Toast.makeText(y.getContext(), text: "No se pudo cerrar sesión. Vuelva a intentarlo.", Toast.LENGTH_LONG).show();
        }
    });
}
```

✓ Permisos:

- Se declararon los permisos necesarios en el archivo **AndroidManifest.xml**.

```
<uses-permission android:name="com.google.android.gms.permission.AD_ID" tools:node="remove"/>
```


VI.I.IV. IMPLEMENTACIÓN DE LA AUTENTICACIÓN CON LA CUENTA DE FACEBOOK A TRAVÉS DE FIREBASE EN UNA APLICACIÓN ANDROID

- ✓ Configuración de Facebook:
 - En **Facebook Developers** básicamente se registraron la app y se obtuvieron el ID de la aplicación y el secreto de la aplicación.
- ✓ Configuración del Inicio de Sesión con Facebook en Firebase:
 - Como ya se mencionó anteriormente, en la Consola de Firebase, se habilitó "Facebook" como método de acceso.
 - Se configuró la integración con Facebook proporcionando el ID de la aplicación y el secreto de la aplicación.
- ✓ Declaración de las dependencias de Facebook:
 - Se agregaron las dependencias en el archivo *build.gradle*.

```
implementation ("com.facebook.android:facebook-login:latest.release")
implementation ("com.facebook.android:facebook-android-sdk:[8,9]")
```

Build.gradle

- ✓ Configuración del Inicio de Sesión con Facebook:
 - Se escribieron el ID de la aplicación de Facebook en **strings.xml** y en el **AndroidManifest.xml**.

```
<string name="facebook_app_id">721168276560335</string>
<string name="fb_login_protocol_scheme">fb721168276560335</string>
<string name="facebook_client_token">791eb4c42e43af0c84ff705d735f6273</string>
```

Strings.xml

```
<activity android:name="com.facebook.FacebookActivity"
    android:configChanges=
        "keyboard|keyboardHidden|screenLayout|screenSize|orientation"
    android:label="Confiax" />
<activity
    android:name="com.facebook.CustomTabActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />
        <data android:scheme="fb721168276560335" />
    </intent-filter>
</activity>
```

AndroidManifest.xml

- ✓ Configuración del Inicio de Sesión con Facebook:
 - Se configuró el botón de inicio de sesión y el **Callbackmanager** en la clase **LoginActivity**.

```
btloginFB = (LoginButton) findViewById(R.id.lbFacebook);
btloginFB.setPermissions(Arrays.asList("email", "user_friends", "user_likes"));
```

```
callbackManager = CallbackManager.Factory.create();
```

```
1
btloginFB.registerCallback(callbackManager, new FacebookCallback<LoginResult>() {
    @Override
    public void onSuccess(LoginResult loginResult) {
        Log.d(TAG, msg: "facebook:onSuccess:" + loginResult);
        handleFacebookAccessToken(loginResult.getAccessToken());
    }
    @Override
    public void onCancel() {
        Log.d(TAG, msg: "facebook:onCancel");
        Toast.makeText(getApplicationContext(), text: "Cancelando acción...", Toast.LENGTH_SHORT).show();
    }
    @Override
    public void onError(@NonNull FacebookException e) {
        Log.d(TAG, msg: "facebook:onError", e);
        Toast.makeText(getApplicationContext(), text: "Error en el inicio de sesión", Toast.LENGTH_SHORT).show();
    }
});
```

✓ Manejo del Token de Acceso de Facebook y Autenticación en Firebase:

- Se creó el método **handleFacebookAccessToken** para manejar el token de acceso de Facebook y autenticar en Firebase.

```
1
btloginFB.registerCallback(callbackManager, new FacebookCallback<LoginResult>() {
    @Override
    public void onSuccess(LoginResult loginResult) {
        Log.d(TAG, msg: "facebook:onSuccess:" + loginResult);
        handleFacebookAccessToken(loginResult.getAccessToken());
    }
    @Override
    public void onCancel() {
        Log.d(TAG, msg: "facebook:onCancel");
        Toast.makeText(getApplicationContext(), text: "Cancelando acción...", Toast.LENGTH_SHORT).show();
    }
    @Override
    public void onError(@NonNull FacebookException e) {
        Log.d(TAG, msg: "facebook:onError", e);
        Toast.makeText(getApplicationContext(), text: "Error en el inicio de sesión", Toast.LENGTH_SHORT).show();
    }
});
```

✓ Manejo del Callback de Facebook en onActivityResult:

- Se implementó el método **onActivityResult** para manejar el resultado del inicio de sesión de Facebook.

✓ Cierre de Sesión con Facebook:

- Para proporcionar una opción de cierre de sesión, se utilizó **LoginManager** y

FirebaseAuth.

✓ Probando la Implementación:

- Se hizo una prueba de ejecución para verificar si el inicio de sesión con Facebook funcionaba correctamente.

VI.I.V. FIRESTORE DATABASE

✓ 1.- Declaración de las Dependencias:

- Se declaró la dependencia necesaria en el archivo build.gradle (módulo de la aplicación).

```
implementation("com.google.firebase:firebase-firestore")
```

✓ 2.- Inicialización de Firestore

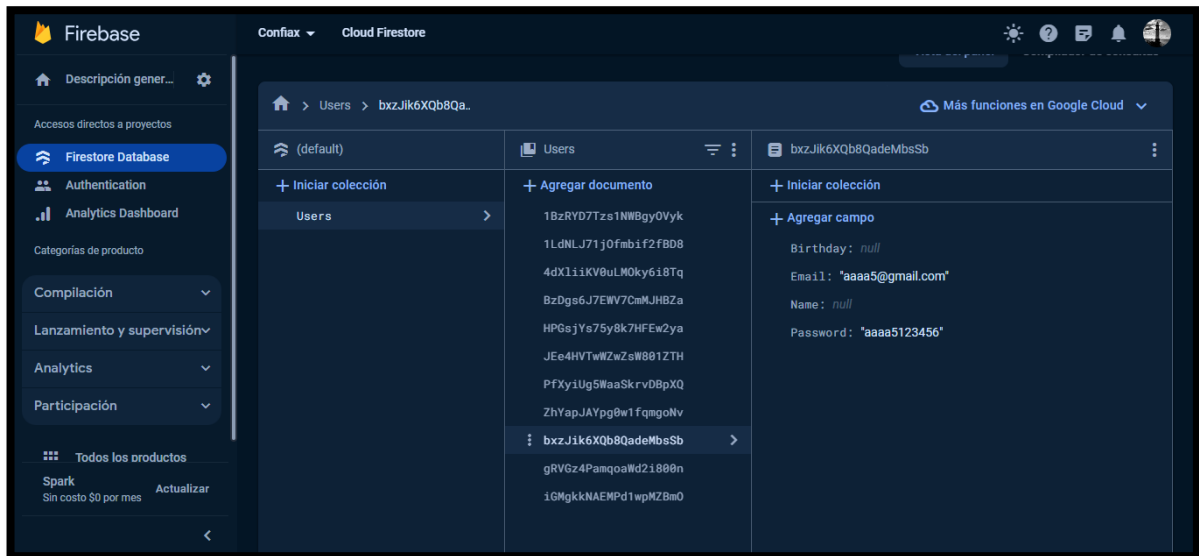
- En la app, se realizó la inicialización de Firestore en la clase **SignUpActivity.java**.

```
private FirebaseFirestore db = FirebaseFirestore.getInstance();
```

✓ 3.- : Registro de los nuevos usuarios en Firestore

- Se creó una colección llamada 'User' en la que se insertaron los usuarios que se registraron por primera vez en la aplicación.
- Al agregar un nuevo documento o registro en Firestore, el sistema generó automáticamente un identificador único para ese documento. De esta forma, se evitaron posibles conflictos con la ID del documento.
- Los campos requeridos para el registro de un usuario son los siguientes:
 - *Id* (tipo String)
 - *Name* (tipo String)
 - *Email* (tipo String)
 - *Password* (tipo String)
 - *Birthday* (tipo String)
- Firestore no tiene una restricción directa para especificar que los campos no deben ser null (es decir, campos obligatorios) en el nivel de la base de datos. Firestore permite documentos con campos nulos. Por ello, se implementan

validaciones en el lado de la aplicación para asegurar que los datos que se intentan almacenar cumplen con ciertos requisitos, como la no nulidad.



```

1 usage
public boolean saveUser (User user) {
    try {
        if (!Objects.isNull(user) || isNull(user)) {
            HashMap hashmap = new HashMap<>();
            hashmap.put("Name", user.getName());
            hashmap.put("Email", user.getEmail());
            hashmap.put("Password", user.getPassword());
            hashmap.put("Birthday", user.getBirthday());

            db.collection( collectionPath: "Users").document().set(hashmap);
        }
    } catch (Exception e){
        /*TODO: pendiente gestionar excepciones*/
        e.printStackTrace();

        return false;
    }
    return true;
}

```

SignUpActivity.java

VI.II. PANTALLAS DE LA APLICACIÓN

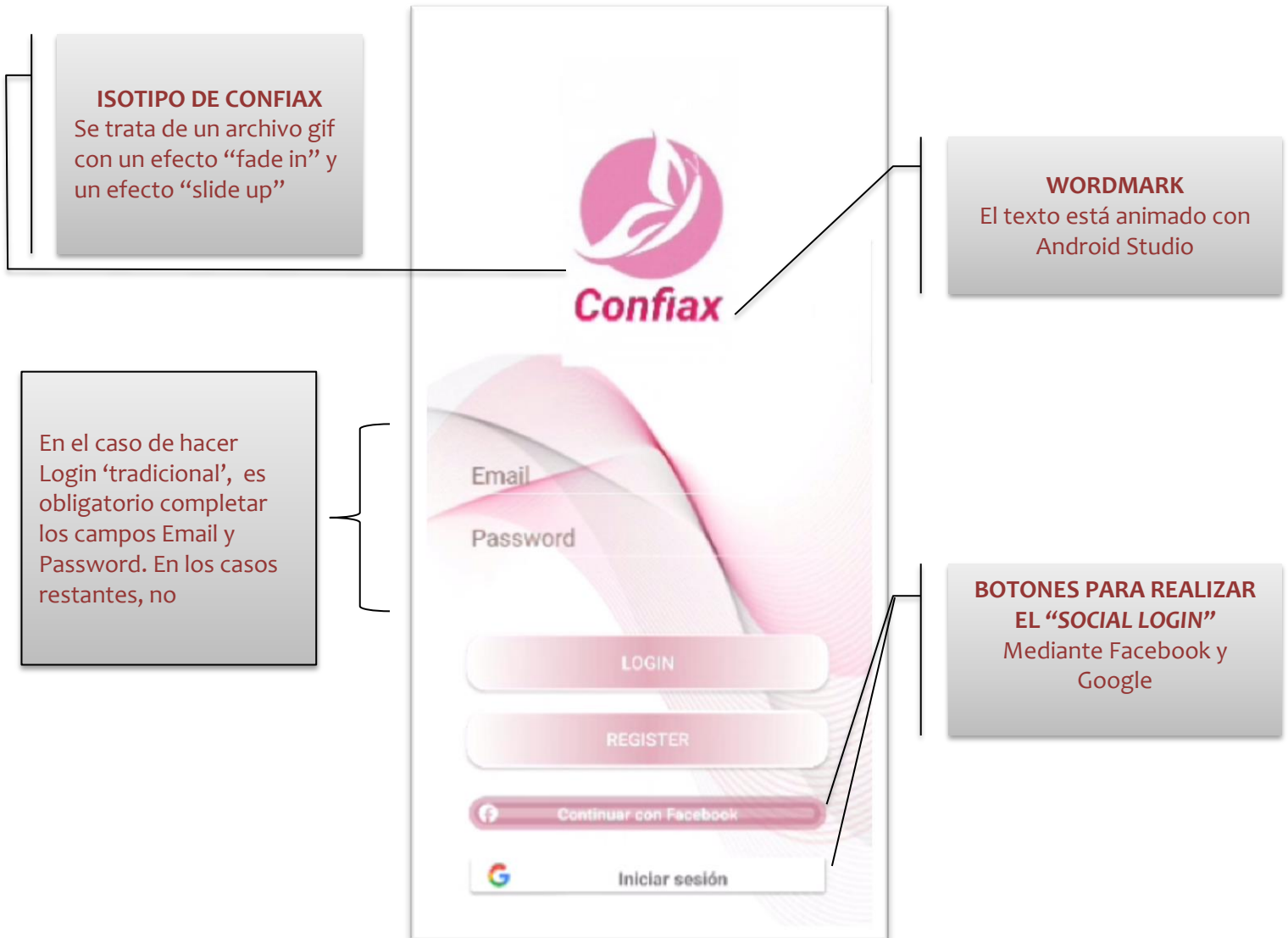
Pantalla de Login

LoginActivity.java

- ✓ En esta pantalla el usuario tiene las siguientes opciones:
- ✓ Registrarse como nuevo usuario pulsando el botón Registrar.
- ✓ Conectarse pulsando el botón “Login” (si ya se ha registrado anteriormente).
- ✓ Conectarse a través de la cuenta personal de Google.
- ✓ Conectarse a través de la cuenta personal de Facebook.



"Login Social" a través de la cuenta Google del usuario



Se trata de la pantalla inicial, la primera que vamos a ver cuando arranquemos la app.

En ella se pide a la usuaria que haga Login. Lo podrá hacer de 3 maneras distintas:

- ✓ Rellenando los campos editables de **Email y Password**.
- ✓ Autenticándose a través de la cuenta de **Google**.
- ✓ Autenticándose a través de la cuenta de **Facebook**.

Cuando la usuaria quiera hacer Login empleando la primera opción (introduciendo email y password) y el sistema no la reconozca o no la encuentre, le mostrará un pop-up informándole de que no está registrada en el sistema y que previamente deberá darse de alta haciendo *click* en el

botón de 'Register'.

Se han implementado una serie de validaciones previas para dar de alta al usuario:

- ✓ No se permite campos vacíos o a null (ambos campos, Email y Password, deben completarse).
- ✓ Se comprueba si el formato del Email es válido. Debe de cumplir el siguiente regex pattern: `^[a-zA-Z0-9.!#$%&'*/+=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$`. La clase **Patterns** de **android.util** ofrece esta misma función, pero de una manera más sencilla, comparando el email introducido con la constante `EMAIL_ADDRESS` que contiene el **regEX** específico para validar emails.
- ✓ Se comprueba si el formato de la Password es válido: debe de tener más de **6 caracteres**, contener al menos **1 letra minúscula**, **1 letra mayúscula**, **1 entero** y **1 carácter especial**.



Una vez la usuaria haya pulsado en 'Register', la dirigirá al [formulario de Alta de Usuario](#).

Pantalla de Alta de Usuario**SignUpActivity.java**

Rellena el formulario

Name

Email

Password

Birthday

SELECT DATE

GUARDAR

Ninguno de los 4 campos puede ser nullable o vacío

BOTÓN PARA SELECCIONAR LA FECHA DE NACIMIENTO
Al seleccionar la fecha, se completa automáticamente el campo de texto de Birthday

Los 4 campos a rellenar para registrarse son obligatorios. Se han implementado una serie de validaciones previas para dar de alta al usuario:

- ✓ No se permite campos vacíos o a null (todos los campos deben de completarse).
- ✓ Se comprueba si el formato del **email** es válido. Debe de cumplir el siguiente regex pattern: `^[a-zA-Z0-9.!#$%&'*/+=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$`. La clase



Patterns de **android.util** ofrece esta misma función, pero de una manera más sencilla, comparando el email introducido con la constante `EMAIL_ADDRESS` que contiene el **regEX** específico para validar emails.

- ✓ Se comprueba si el formato de la Password es válido: debe de tener más de **6 caracteres**, contener al menos **1 letra minúscula**, **1 letra mayúscula**, **1 entero** y **1 carácter especial**.

- ✓ La **fecha de nacimiento** es tratada como una cadena de texto y no hace falta validar su formato, pues la obtiene debidamente formateada del **CalendarView** ("dd/MM/yy"). Al lado del campo editable hay un botón ("Seleccionar") que al presionar muestra el calendario para seleccionar, sin teclear, la fecha de nacimiento.

Pantalla Principal

MenstrualCalendarActivity.java

Leyenda Calendario

COMPONENTE

CUSTOMCALENDARVIEW de la librería homónima de *Staktips*.

Es un componente basado en *CalendarView*, programable, que ofrece más opciones de configuración que el componente de la librería de Android. Arriba a la derecha, se muestra la leyenda de colores de cada uno de los eventos

BOTÓN DE DESCONEXIÓN

Saca al suario de la aplicación y lo lleva a la pantalla de Login

NOV 2023

lun	mar	mié	jue	vie	sáb	dom
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

PREFERENCIAS

Menstruación

Fertilidad

Ovulación

BOTÓN DE PREFERENCIAS

Abre la pantalla de configuración en la que la usuaria mete las variables necesarias para hacer el cálculo matemático que estima la fecha de la próxima menstruación, ovulación y periodo fértil. Estas variables son:

- ✓ Duración de la regla
- ✓ Periodicidad (medida en días) del inicio de cada regla
- ✓ Fecha de la última regla

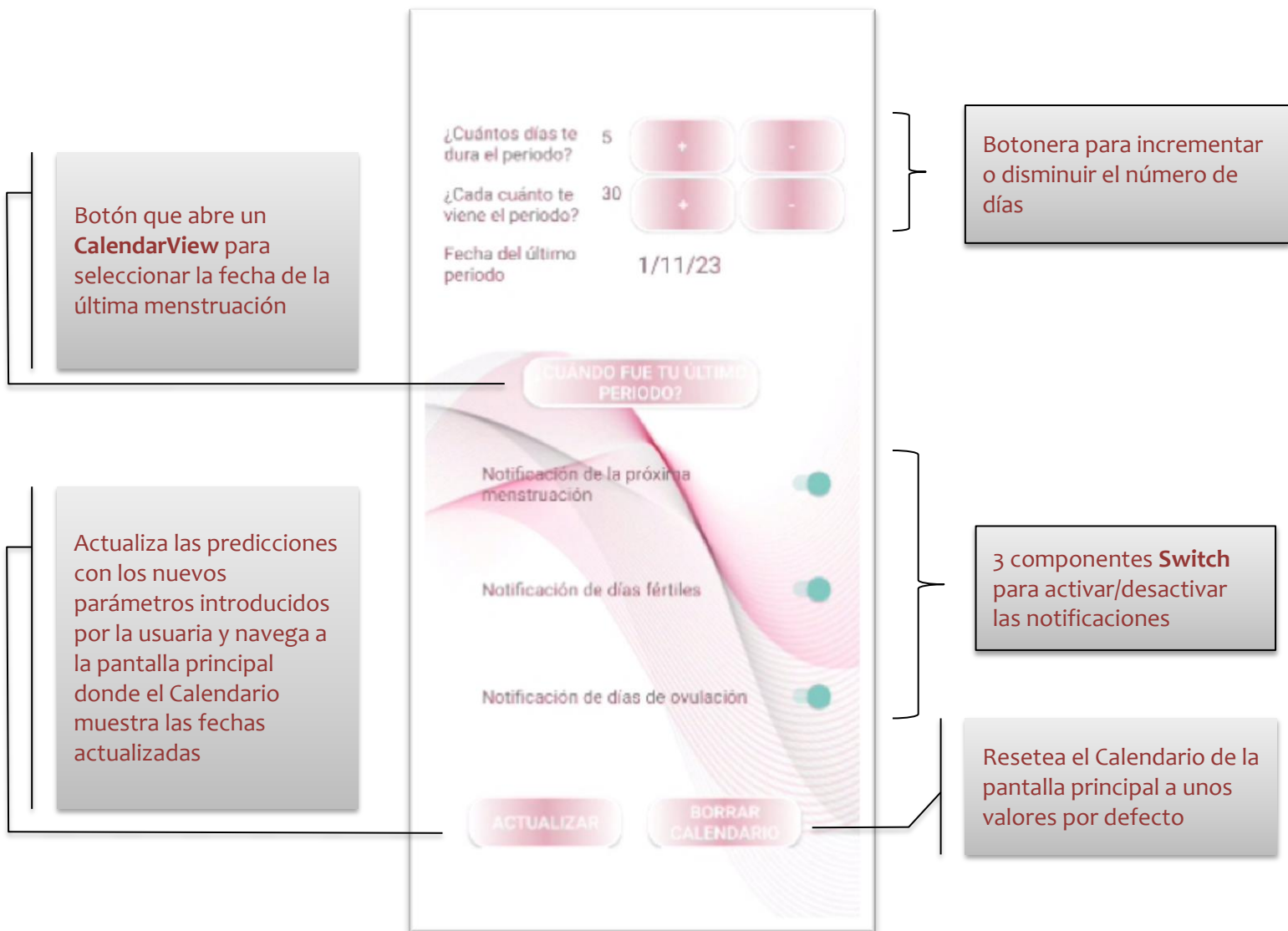
Una vez logueados o registrados en la app, se llega a la pantalla principal. Aquí la usuaria debe pulsar el botón de '*Preferencias*' para meter una serie de datos obligatorios para calcular las fechas de la próxima menstruación/ovulación/periodo de fertilidad y para poder ver los resultados reflejados en el calendario. Cada uno de esos eventos estará diferenciado del resto por un color distintivo.

Desde el principio, se quería usar un calendario que fuera personalizable, más allá de las posibilidades que ofrece el componente **CalendarView** de la librería estándar de Android. De entre

todas las librerías candidatas, se eligió **CustomCalendarView** de la librería de **Stacktips** (<https://stacktips.com/articles/custom-calendar-view-library-in-android>) por su fácil implementación. Da la posibilidad de ajustar varios aspectos visuales, como el color para el fondo, para el día, la semana y el título del calendario. Permite controlar la visualización de los días que se desbordan en el mes siguiente y anterior. Da la posibilidad de establecer opciones personalizadas para el día de la semana en el que comienza el calendario. De forma predeterminada, está configurado en `Calendar.SUNDAY` (domingo), pero para la app se ha establecido que comience el Lunes. También permite especificar un tipo de letra personalizado utilizando el método **setCustomTypeFace()**. En definitiva, a la usuaria le resultará muy intuitivo debido a su sencillez en el diseño y manejo: enseguida sabrá navegar por el calendario e interpretar los datos y valores que en él se reflejarán.

Pantalla Preferencias o de Configuración

PreferencesActivity.java



Se trata de la pantalla de configuración en la que el usuario introduce una serie de variables para hacer operaciones con el algoritmo que predice el próximo periodo.

Aunque dichas fórmulas matemáticas no entrañen una enorme complejidad, la dificultad se presenta a la hora de gestionar la entrada de datos y, en base a ella, recalcular con cada actualización.

```

1 usage
private List<LocalDate> getPeriod(LocalDate periodStart) {
    List<LocalDate> period = new ArrayList<>();
    LocalDate periodDay = new LocalDate(periodStart);
    period.add(periodDay);

    for (int i = 0; i < manager.getMenstruationLength() - 1; ++i) {
        periodDay = periodDay.plusDays(1);
        period.add(periodDay);
    }

    return period;
}

1 usage
private List<LocalDate> getFertile(LocalDate periodStart) {
    int diff = manager.getPeriodLength() - 20;
    LocalDate fertileDay = periodStart.plusDays(diff);
    List<LocalDate> fertile = new ArrayList<>();
    fertile.add(fertileDay);

    for (int i = 0; i < 6; ++i) {
        fertileDay = fertileDay.plusDays(1);
        fertile.add(fertileDay);
    }

    return fertile;
}

```

MenstrualCycleCalculator.java

```
1 usage
public void updatePreferences(View view) {
    SharedPreferences preferences = getSharedPreferences(AppPreferences.SHARED_PREFERENCES_FILE, MODE_PRIVATE);
    boolean userPreferencesAvailable = preferences
        .getBoolean(AppPreferences.BASIC_USER_PREFERENCES_AVAILABLE, defValue: false);
    SharedPreferences.Editor editor = preferences.edit();
    savePreferenceStringValue(R.id.menstruation_length_value, AppPreferences.MENSTRUATION_LENGTH_KEY, editor);
    savePreferenceStringValue(R.id.period_length_value, AppPreferences.PERIOD_LENGTH_KEY, editor);
    savePreferenceStringValue(R.id.last_period_date_value, AppPreferences.LAST_PERIOD_DATE_KEY, editor);
    savePreferenceBooleanValue(R.id.incoming_period_notification,
        AppPreferences.INCOMING_PERIOD_NOTIFICATION_KEY, editor);
    savePreferenceBooleanValue(R.id.fertile_days_notification,
        AppPreferences.FERTILE_DAYS_NOTIFICATION_KEY, editor);
    savePreferenceBooleanValue(R.id.ovulation_notification,
        AppPreferences.OVULATION_NOTIFICATION_KEY, editor);

    editor.putBoolean(AppPreferences.BASIC_USER_PREFERENCES_AVAILABLE, true);
    editor.apply();

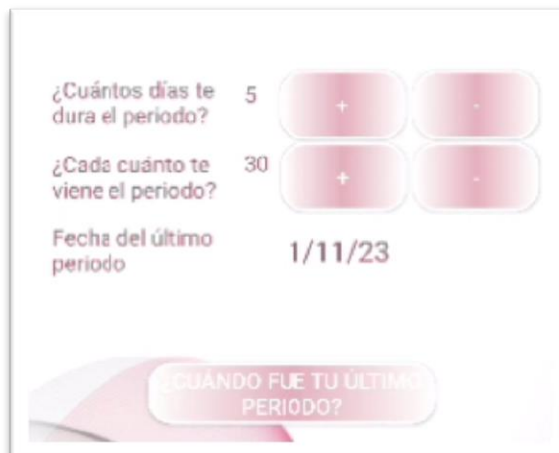
    if (!userPreferencesAvailable) {
        recalculateReceiver.setPredictionService(this);
    }

    new MenstrualCycleCalculator(new MenstrualCycleManager( cbc: this)).calculate();
    finish();
}
```

PreferencesActivity.java

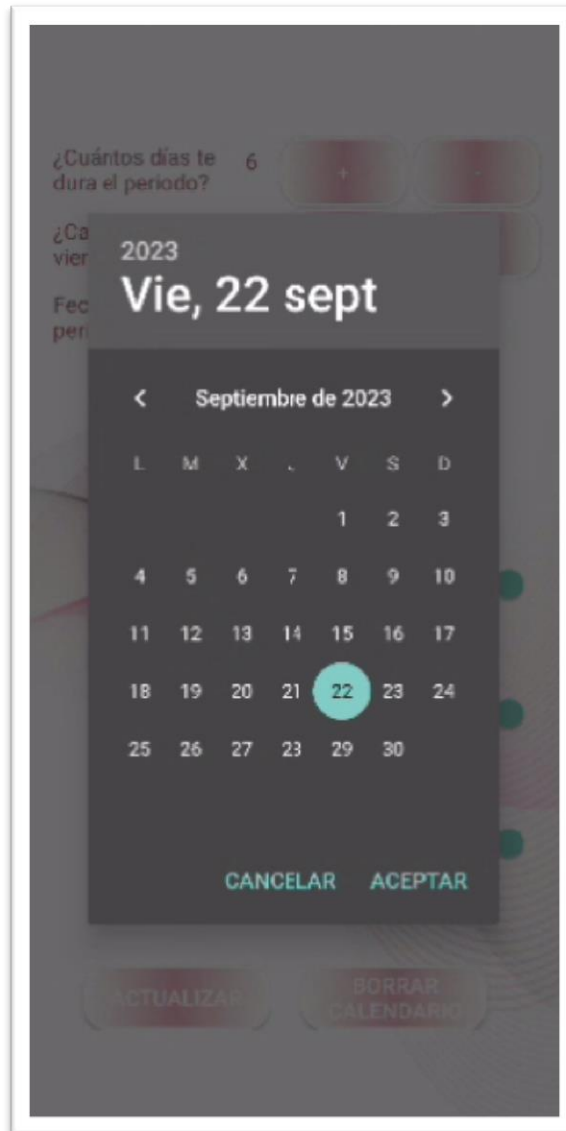
Como se ha comentado antes, para predecir las próximas fechas de la menstruación, ovulación y días de mayor fertilidad, la usuaria debe de introducir:

- ✓ Duración del periodo.
- ✓ Cada cuántos días le viene.
- ✓ Fecha del inicio del último periodo.



- La usuaria puede incrementar o disminuir las cifras con la botonera que hay a la derecha de la pantalla.

- Para la duración de la menstruación, el rango de valores está entre 1 y 12 días.
- Para la periodicidad de la menstruación, el rango está entre 18 y 60 días.
- En cuanto a la fecha de la última menstruación, el usuario debe pulsar el botón '*¿Cuándo fue tu último periodo?*' y en ese momento salta una notificación en la que se informa o recuerda a la usuaria de que debe de introducir el día del inicio de la regla.



CalendarView para seleccionar la fecha de la última menstruación



Selección de *Switch* para activar/desactivar las notificaciones

- En la mitad inferior de la pantalla hay tres componentes **Switch** que habilita/deshabilita la función de notificar a la usuaria de cada uno de los tres eventos que en la app se tratan: *menstruación/ovulación/fertilidad*.
- Cada Switch funciona de manera independiente.
- Cuando la usuaria termine de meter toda la información, pulsará el botón '*Actualizar*' y, a continuación, navegará a la pantalla principal (**MenstrualCalendarActivity.java**) donde podrá consultar los resultados de lo que ha introducido y seleccionado en la pantalla de '*Preferencias*'.



Estimación de la siguiente menstruación/ovulación/días fértiles de una usuaria cuya última regla le vino el 22/09/2023

VI.III. REQUISITOS

Se define la lista de requisitos funcionales que deberá cubrir la aplicación:

1. Usuario

R1. Creación de un usuario.

La aplicación permite crear un usuario.

R2. Inicio de sesión.

La aplicación permite iniciar sesión a un usuario.

R3. Cierre de sesión.

La aplicación permite cerrar sesión a un usuario.

2. Registro menstruación

R4. Creación de un registro.

La aplicación permite crear registros de la menstruación.

R5. Actualización de registro.

La aplicación permite modificar registros de la menstruación.

R6. Borrado de registro.

La aplicación permite borrar registros de la menstruación.

R7. Filtrado de notificaciones.

La aplicación permite filtrar las notificaciones de la menstruación/ovulación/días fértiles.

VI.IV. CASOS DE USO

CU1 .- Un usuario se registra en el sistema

Requisitos: R1

Descripción:

- El usuario indica Email y Password.
- Si ese usuario no está dado de alta en el sistema, se navega a la pantalla de Alta de Usuario.
- El usuario deberá de rellenar los campos de Name, Email, Password y Birthday.
- Para finalizar se selecciona la opción de “Guardar”.

CU2.- Un usuario inicia sesión

Requisitos: R2

Descripción:

- El usuario indica Name y Password.
- Si el usuario ya se ha dado de alta en el sistema anteriormente, se mostrará un pop-up de que ha accedido con éxito a la aplicación.
- La aplicación manda al usuario a la Home.

CU3 .- Un usuario cierra sesión

Requisitos: R3

Descripción:

- El usuario accede a la Home de la aplicación.
- Selecciona la opción de “SignOut” →

- La aplicación manda al usuario a la pantalla de Login.

CU4.- Un usuario crea un registro

Requisitos: R4

Descripción:

- El usuario accede a la pantalla de Preferencias.
- El usuario selecciona el botón +/- para añadir los días de duración del periodo.
- Selecciona el botón +/- para añadir la periodicidad de la menstruación (en días).
- Selecciona el botón “¿Cuándo fue su último periodo?”.
- Selecciona la fecha del inicio de su última regla en el calendario.
- Selecciona la opción de Notificación de la próxima menstruación.
- Selecciona la opción de Notificación de la próxima ovulación.
- Selecciona la opción de Notificación del próximo periodo fértil.
- Selecciona el botón “Actualizar”.
- La aplicación manda al usuario a la Home.

CU5.- Un usuario actualiza registro

Requisitos: R5

Descripción:

- El usuario accede a la pantalla de la Home.
- El usuario selecciona el botón “Preferencias”.
- El usuario accede a la pantalla de Preferencias.

- El usuario selecciona el botón +/- para añadir los días de duración del periodo.
- Selecciona el botón +/- para añadir la periodicidad de la menstruación (en días).
- Selecciona el botón “¿Cuándo fue su último periodo?”.
- Selecciona la fecha del inicio de su última regla en el calendario.
- Selecciona el botón “Actualizar”.
- La aplicación manda al usuario a la Home.

CU6.- Un usuario elimina registro

Requisitos: R6

Descripción:

- El usuario accede a la pantalla de la Home.
- El usuario selecciona el botón “Preferencias”.
- El usuario accede a la pantalla de Preferencias.
- Selecciona el botón “Borrar Calendario”.
- La aplicación manda al usuario a la Home.

CU7.- Un usuario filtra las notificaciones

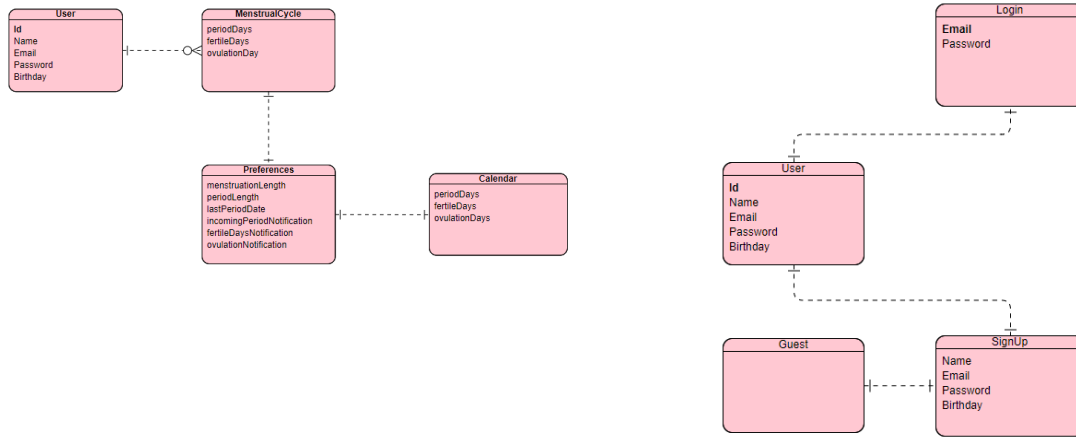
Requisitos: R7

Descripción:

- El usuario accede a la pantalla de la Home.
- El usuario selecciona el botón “Preferencias”.
- El usuario accede a la pantalla de Preferencias.

- Selecciona la opción de Notificación de la próxima menstruación.
- Selecciona la opción de Notificación de la próxima ovulación.
- Selecciona la opción de Notificación del próximo periodo fértil.
- Selecciona el botón “Actualizar”.
- La aplicación manda al usuario a la Home.

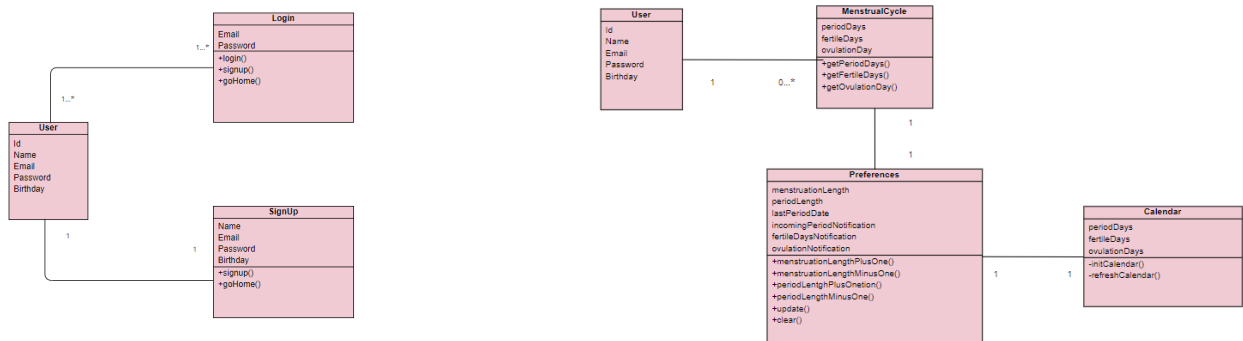
VI.V. DIAGRAMA DE E/R:



Proceso de Autenticación
y de Alta de Usuario en el
sistema

Proceso de creación, actualización y
borrado de registro de
menstruaciones.

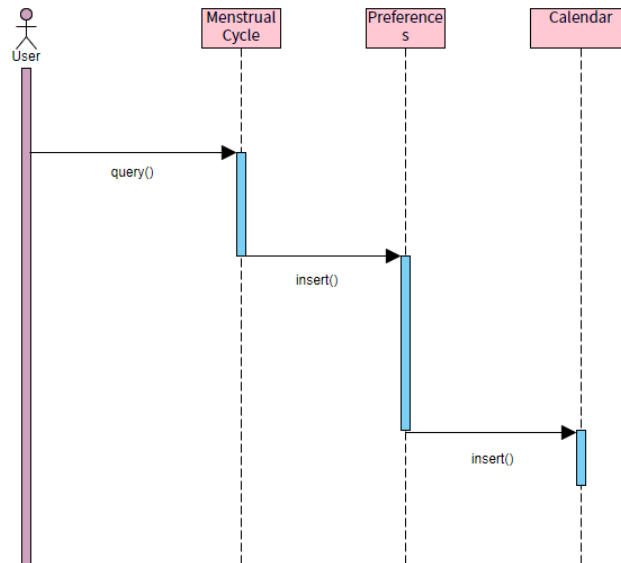
VI.VI. DIAGRAMA DE CLASES:



Proceso de Autenticación
y de Alta de Usuario en el
sistema

Proceso de creación, actualización y
borrado de registro de
menstruaciones.

VI.VII. DIAGRAMA DE SECUENCIA:



Proceso de creación, actualización y borrado de registro de menstruaciones.

VII. CONCLUSIONES

Como conclusión, en lo personal ha sido un reto enfrentarme al diseño y desarrollo de una app para móviles, pues es el primero que hago y, además, con poco margen de tiempo. Gracias a las nociones adquiridas en la asignatura de Programación multimedia y dispositivos móviles he podido saber por dónde empezar y eso no es poco. La herramienta Android Studio y su paradigma de programación es terreno desconocido para mí. Sin embargo, la programación en Java no, y eso me ha dado bastante ventaja. El siguiente paso, es seguir creando apps para dispositivos móviles con *Kotlin* que, por lo que he leído en foros especializados, es mejor opción para aplicaciones destinadas al sistema operativo Android y, además, está bastante demandado en el mercado laboral de la programación.

He quedado bastante contenta con el resultado de **Confiax**. Soy consciente de que no es una propuesta innovadora y que hay incontables apps que hacen el seguimiento del ciclo menstrual, estudios estadísticos, informes personalizados e, incluso, implementan otras funcionalidades que no están relacionadas con el ámbito de la ginecología. Sin embargo, considero que ha sido un acierto el haber elegido la temática de la salud sexual femenina para crear mi primera app para sistemas Android. Considero que este tipo de aplicaciones ofrece una solución integral y personalizada para las usuarias que desean llevar un seguimiento preciso de su ciclo menstrual. Facilita el registro de eventos menstruales, proporcionando una visión clara y detallada del historial menstrual. La capacidad de monitorizar patrones y tendencias a lo largo del tiempo brinda a las usuarias una mayor comprensión de su salud reproductiva.

La función de notificación juega un papel crucial al alertar a las usuarias sobre fechas clave, como el inicio de la menstruación, la ovulación y otros eventos relevantes. Este aspecto permite, sin duda, una mejor planificación y gestión del bienestar menstrual. Precisamente es uno de los puntos que no he conseguido rematar en la app y que considero como una mejora pendiente de implementar. He estado investigando el por qué no funciona dicho servicio, pero no he llegado a nada concluyente. A la hora de ejecutar la app, no se muestra ningún error o *warning* reseñable que me dé una pista. Considero que este sí sería un punto importante a corregir, no sólo porque ya está codificado y no funciona, sino porque la app pierde gran parte de su verdadera función útil, que es la de recordar a la usuaria que se aproximan las fechas de menstruación, ovulación o periodo de mayor fertilidad.

También quedó por añadir la **leyenda del calendario menstrual**. Los códigos de colores los dejo indicados en [este apartado](#). La idea era que cuando la usuaria posicionara el puntero del ratón sobre los días marcados o coloreados del calendario, se mostrara un *tooltip* con la información del tipo de evento. De hecho, hice el intento de implementarlo hasta el último momento, pero el resultado más ‘válido’ no dejaba que la usuaria interactuase con la pantalla porque se quedaba ‘congelada’.

Otra funcionalidad que barajé implementar, fue la de **exportar datos y compartir información con profesionales de la salud** para completar y facilitar información más pormenorizada de los ciclos menstruales.

En resumen, este tipo de aplicaciones no sólo simplifica el seguimiento del ciclo menstrual, sino que también es un servicio de apoyo para las usuarias al proporcionar información valiosa para tomar decisiones informadas sobre su salud reproductiva. Este enfoque integral y centrado en el usuario refleja su compromiso con el bienestar y la autonomía de las mujeres en todas las etapas de su vida.

VIII. CITACIONES Y BIBLIOGRAFÍA

<https://hellocloe.com/es/articulos/ciclo-menstrual>

<https://hellocloe.com/es/articulos/ciclo-a-z/que-es-lo-normal-duracion-y-variacion-del-ciclo-menstrual>

<https://www.womanlog.com/es/>

<https://flo.health/es/herramientas>

<https://www.natalben.com/calculadora-de-la-ovulacion>

<https://espanol.womenshealth.gov/ovulation-calculator>

<https://www.evaxtampax.es/es-es/evax/calendario-menstrual/>

<https://eticasfoundation.org/es/femtech-mi-cuerpo-mis-datos-sus-normas/>

- <https://github.com/>:

<https://github.com/pub-calculator-io/period-calculator>

<https://github.com/anangkur/medi-ku>

<https://github.com/StackTipsLab/custom-calendar-view/blob/master/library/src/com/stacktips/view/CustomCalendarView.java>

- <https://developer.android.com/>:

<https://developer.android.com/studio/intro?hl=es-419>

<https://developer.android.com/studio/write/java8-support?hl=es-419>

- Programación multimedia y dispositivos móviles (TSDAM) > Contenidos:

<https://campusccc.instructure.com/courses/3215/modules>



IX. ANEXOS

El proyecto está publicado en Github: https://github.com/Mallusky/Confiax_ccc