



CONFIAX

CICLO: TS Desarrollo de Aplicaciones Multiplataforma

Autor: Ana Malla Espinar

Tutor: Alberto Castedo Espeso

Curso académico 2023-2024

I. Resumen

Confiar 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. 1
II.	<u>Índice</u>	págs. 2 y 3
III.	<u>Justificación del proyecto</u>	págs. 4 y 5
IV.	<u>Introducción</u>	pág. 5
V.	<u>Objetivos</u>	pág. 6
VI.	<u>Desarrollo</u>	págs. 6 - 23
VI.I.	<u>Tecnologías empleadas</u>	pág. 6
VI.I.I.	<u>Implementación de la BBDD</u>	págs. 7 - 9
VI.I.I.I.	<u>Firestore Database</u>	pág. 7
VI.I.I. II.	<u>Shared Preferences</u>	pág. 9
VI.I. II.	<u>Servicio de autenticación a través de Firebase</u>	págs. 9 y 10
VI.I.III.	<u>Implementación de la autenticación con la cuenta de Google a través de Firebase en una aplicación Android</u>	págs 10 y 11
VI.I. IV.	<u>Implementación de la autenticación con la cuenta de Facebook a través de Firebase en una aplicación Android</u>	págs. 11 y 12
VI.II.	<u>Pantallas de la aplicación</u>	págs. 13 - 18
VI.III.	<u>Requisitos</u>	págs. 18 y 19
VI.IV.	<u>Casos de uso</u>	págs. 19 - 22
VI.V.	<u>Diagrama de E/R</u>	pág. 22
VI.VI.	<u>Diagrama de clases</u>	pág. 23
VI.VII.	<u>Diagrama de secuencia</u>	pág. 23

VII. Conclusiones pág.s. 23 y 24

VIII. Citaciones y Bibliografía pág. 24

IX. Anexos pág. 24

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 en 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 está marcada por un ritmo acelerado que dificulta a la mujer el cuidado de su salud ginecológica y sexual. Detectar alteraciones tales como cambios en la duración del ciclo menstrual, es fundamental para la detección temprana de posibles patologías.

En este contexto, el uso de aplicaciones que ofrecen servicios de registro y consulta menstrual, así como cálculos de la fase de ovulación y fertilidad, simplifica este seguimiento. Además, estas aplicaciones presentan ventajas económicas y ambientales y comodidad sobre los métodos tradicionales, como el uso de papel, agendas, etc.

El mercado actual ofrece una amplia variedad de estas aplicaciones, como [WomanLog](#), [Clue](#) y [Flo](#), que permiten un seguimiento detallado del ciclo menstrual, predicciones de ovulación y fertilidad, notificaciones y recordatorios útiles, personalización de la experiencia del usuario y enfoque en la privacidad y seguridad de los datos.

Confiax, basándose en estas apps, implementa una serie de algoritmos para calcular los periodos menstruales, de ovulación y de fertilidad, ofrece servicio de notificaciones o alertas y, además, permite a las usuarias introducir sus propios datos menstruales para obtener resultados personalizados y precisos.

En conclusión, **Confiax** pretende ser una herramienta integral para el monitoreo y la gestión de la salud menstrual de la mujer y pone énfasis en la usabilidad y precisión de los datos, con un diseño atractivo, funciones intuitivas y fácil disponibilidad en la plataforma Android.

(© 2024 womanlog.com, s.f.): <https://play.google.com/store/apps/details?id=com.womanlog&pli=1>

(© 2024 Clue de Biowink GmbH, s.f.): https://play.google.com/store/apps/details?id=com.clue.android&pcampaignid=web_share

(© 2024 Flo Health Inc., s.f.): https://play.google.com/store/apps/details?id=org.iggymedia.periodtracker&pcampaignid=web_share

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, en programación Java y en el desarrollo de interfaces gráficas en dispositivos móviles.

VI. Desarrollo

VI.I. **Tecnologías empleadas**

Para el desarrollo de la app se han empleado las siguientes tecnologías:

- Lenguaje de programación Java versión 17.
- IDE Android Studio Giraffe | 2022.3.1 Patch 4.
- Gradle versión 8 (<https://docs.gradle.org/8.0/release-notes.html>).
- SDK de Android versión 34.
- Librería CustomCalendarView, que utiliza la SDK versión 27

VI.I.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.

Se instaló el producto (**MongoDB**), la herramienta de la Shell (**MongoShell**) y el software de visualización de la BBDD (**MongoDBCompass**), y se creó un clúster de BBDD en **MongoDB Atlas**. 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 [Firebase de Google](#). Se siguieron los siguientes pasos:

- Se dio de alta la app **Confiax** en Firebase.
- Se descargó y agregó el archivo de configuración *google-services.json* al proyecto (en el directorio 'app').
- Se añadieron las dependencias de Firebase en el archivo *build.gradle* del módulo de la aplicación, para habilitar los servicios necesarios (Figura 1).

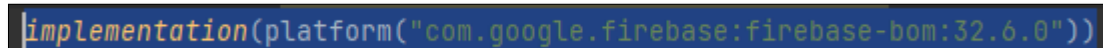


Figura 1: Declaración de la dependencia de la librería Firebase en el archivo de configuración *build.gradle*.

- 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.

VI.I.I.I.I. Firestore database

Se siguieron los siguientes pasos para su configuración:

- **Declaración de las Dependencias:** se declaró la dependencia necesaria en el archivo *build.gradle* del módulo de la aplicación (Figura 2).


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

Figura 2: Declaración de la dependencia de la librería Firebase-Firestore en el archivo de configuración build.gradle, a nivel de módulo de la aplicación.

- Inicialización de **Firestore** en la clase **SignUpActivity.java**.
- **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 evitan posibles conflictos con la ID del documento (Figura 3).
 - 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)

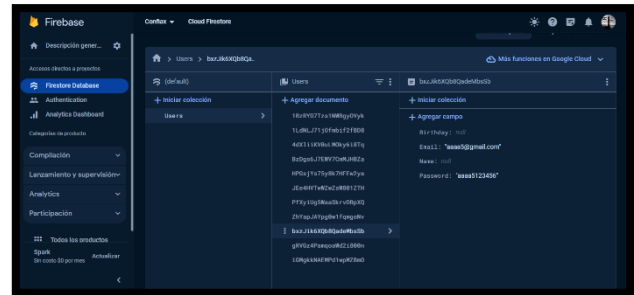


Figura 3: Colección “User” en la BBDD Firestore.

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

VI.I.I. II. Shared Preferences

- **Shared Preferences** es un mecanismo de almacenamiento ligero para datos clave-valor en Android. Se utiliza para almacenar pequeñas cantidades de datos primitivos, como cadenas, enteros, booleanos y conjuntos de cadenas, que son específicos de una aplicación.

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
  <boolean name="com.ccc.confiax.fertile_days_notification" value="false" />
  <boolean name="com.ccc.confiax.incoming_period_notification" value="true" />
  <string name="com.ccc.confiax.menstruation_length">5</string>
  <boolean name="com.ccc.confiax.ovulation_notification" value="true" />
  <boolean name="basic_user_preferences_available" value="true" />
  <string name="com.ccc.confiax.period_length">28</string>
  <string name="com.ccc.confiax.last_period_date">1/6/24</string>
  <string name="com.ccc.confiax.period_days_beans_list">{"earliestDate":"2024-04-04","fertileDays":["2024-04-13",
</map>
```

Contiene pares clave-valor que almacenan las preferencias de la usuaria y los datos de su ciclo menstrual.

VI.I. II. Servicio de autenticación a través de Firebase

Para la implantación del proceso de autenticación a través de Firebase, se siguieron los siguientes procedimientos:

- **Se agregó la dependencia de Firebase Authentication:** En el archivo *build.gradle* (módulo de la aplicación), se declaró la dependencia para Firebase Authentication (Figura 4).

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

Figura 4: Declaración de la dependencia de la librería *Firebase-auth* en el archivo de configuración *build.gradle* a nivel de módulo de la aplicación.

- **Se configuraron los proveedores de autenticación:** En la Consola de Firebase → sección de "Authentication" → pestaña "Método de acceso". Se configuraron dos proveedores de autenticación: **Google** y **Facebook** (Figura 5).

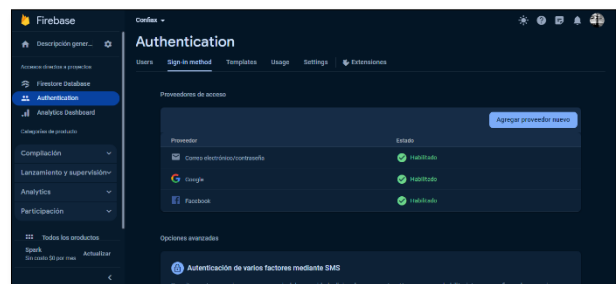


Figura 5: Menú de Autenticación en la consola de Firebase.

- Se implementó el código correspondiente para el manejo de sesiones y cierre de sesión (Figura 6).

```

//login
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()) {
                //login()
                // Sign in success, update UI with the signed-in user's information
                Log.d(TAG, "signInWithEmail:success");
                FirebaseAuth user = FirebaseAuth.getCurrentUser();
                updateUser(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();
            }
        }
    });
}

```

Figura 6: Implementación de la autenticación en la aplicación: método *login* de la clase *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 **Confiax**.
 - 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.confiax** en el archivo *build.gradle*).
 - 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 (Figura7):

```

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: 
SHA1: 
SHA-256: 
Valid until: jueves, 30 de octubre de 2053

```

Figura 7: Tarea ejecutada en consola para obtener certificación de la clave de firma de la aplicación.

- **Configuración en Firebase:**
 - En la Consola de Firebase se seleccionó el proyecto **Confiax**.

- En la sección de “Autenticación”, en la pestaña “Método de acceso”, se habilitó “Google” como método de acceso y se introdujo el ID del cliente de OAuth.
- **Declaración de dependencia de Firebase Auth:**
 - Se añadió la dependencia de Firebase Authentication en el archivo *build.gradle*.
- **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.
- **Autenticación en Firebase:**
 - Se implementó el método **firebaseAuthWithGoogle()** para autenticar con Firebase.
- **Manejo del Cierre de Sesión:**
 - Se hace uso del método “*signOut*” de la clase **GoogleSignInClient** para desconectar al usuario de Google y, luego, se llama a “*firebaseAuth.signOut()*” para cerrar sesión en Firebase.
- **Permisos:**
 - Se declaró, entre otros permisos de usuario, uno para cargar anuncios con fines no publicitarios para el análisis de datos por parte de la plataforma Firebase (“*firebase-analytics*”) en el archivo *AndroidManifest.xml* (Figura 8). De lo contrario, se generaría un error en la aplicación.

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

Figura 8: Declaración de permisos de usuario para gestión de anuncios por parte de Firebase Analytics.

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* (Figura 9).

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

Figura 9: Declaraciones de las dependencias de las librerías Facebook-login y Facebook-android-sdk en el archivo de configuración *build.gradle* a nivel de módulo de la aplicación.

- **Configuración del Inicio de Sesión con Facebook:**

- Se escribieron el ID de la aplicación de Facebook en *strings.xml* y en el *AndroidManifest.xml*.
- Se configuró el botón de inicio de sesión y el **Callbackmanager** en la clase **LoginActivity**.

- **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.

- **Manejo del Callback de Facebook en método 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.II. Pantallas de la aplicación

Imagotipo (Figura 10)

A continuación, se explica de manera resumida las razones por las que se ha elegido este imagotipo y los colores corporativos para representar a la app



Confiax:

Figura 10: Isotipo de la app Confiax

- **La Mariposa:** Simboliza transformación, renovación, feminidad, delicadeza, libertad y esperanza.
- **El color rosa:** Se asocia con lo femenino, el amor, el cuidado, la tranquilidad y la relajación.

La combinación de ambos elementos da como resultado un mensaje **positivo, empoderador** y de **confianza** (de ahí el origen del nombre de esta app) sobre el ciclo menstrual.

Pantalla de Login (Figura 11)

LoginActivity.java

Se trata de la pantalla inicial, la primera que vamos a ver cuándo arranquemos la app. En ella, el usuario tiene las siguientes opciones:

- Registrarse como nuevo usuario pulsando el botón “*Register*”.
- 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.

La usuaria podrá hacer *login* 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.

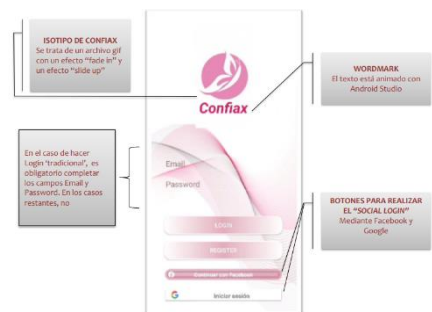


Figura 11: Esquema de la pantalla de Login

Cuando la usuaria introduzca *Email* y *Password* y el sistema no la reconozca o no la encuentre, se 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 “*Register*”.

Para ello, se han implementado una serie de validaciones previas para dar de alta al usuario:

- **No se permiten 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.



Figura 11: Validaciones en la acción de Login

- **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 (Figura 12)

SignUpActivity.java

Los 4 campos que 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.!#$%&'*/+=?^_`{|}~]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$`.
- **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

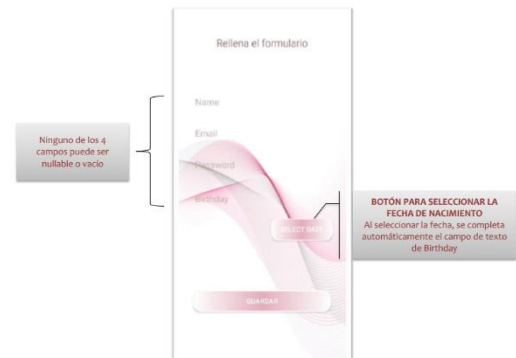


Figura 12: Esquema de la pantalla de Alta de Usuario



Figura 12: Validaciones en la acción de Alta de Usuario

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 (Figura 13)

MenstrualCalendarActivity.java

Una vez se ha iniciado sesión o el usuario se ha registrado 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.



Figura 13: Esquema de la pantalla del Calendario Menstrual

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.



Figura 12: Pantalla del Calendario Menstrual (versión definitiva)

Pantalla Preferencias o de Configuración (Figura 14)

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, recalculan con cada actualización.

Como se ha comentado antes, para predecir las próximas fechas de la menstruación, ovulación y días de mayor fertilidad, la usuaria debe de introducir en el cuadro de control (Figura 15)



Figura 14: Pantalla de entrada de datos de la usuaria (pantalla de Preferencias de usuaria).

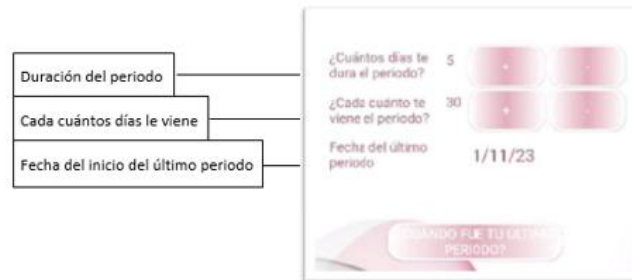


Figura 15: Cuadro de control donde se introducen los parámetros de entrada para calcular próxima menstruación y las próximas fechas de ovulación y de días fértiles.

Funcionamiento del cuadro de control (Figura 16)

- 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.

- 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'.

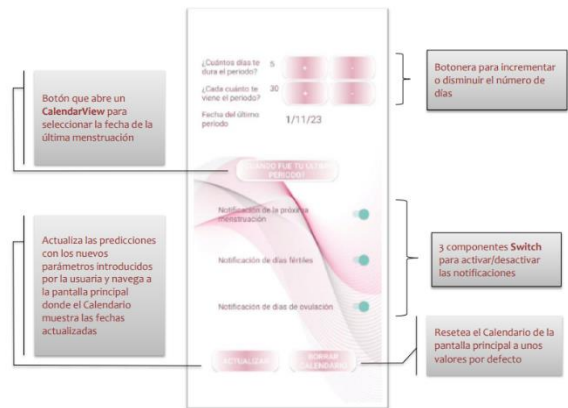


Figura 14: Esquema de la pantalla de Preferencias de usuaria.

Parámetro E/	Descripción	Tipo de dato	Valor inicial (días)	Rango permitido (días)	Botones	Control adicional
Duración de la menstruación	Días que dura la menstruación	Numérico	1	1 - 12	↑, ↓	-
Periodicidad de la menstruación	Días que transcurren entre el inicio de una menstruación y el inicio de la siguiente	Numérico	18	18 - 60	↑, ↓	-
Fecha de la última menstruación	Fecha en la que comenzó la última menstruación	Fecha (dd/MM/aa)	-	-	"¿Cuándo fue tu último periodo?"	Calendario

Figura 16: Tabla de la información que va a mostrar el Cuadro de Mandos

Ejemplo de uso del cuadro de control de ciclo menstrual:

- Situación inicial:** La usuaria abre la aplicación y se dirige a la pantalla de Preferencias con los valores iniciales (Figura 17)

Parámetro E/	Valor
Duración de la menstruación	5 días
Periodicidad de la menstruación	29 días
Fecha de la última menstruación	04/04/2024

Tabla de los datos registrados de la última menstruación de la usuaria: la duración, la periodicidad y la fecha de la última regla.



Figura 17: Pantalla de Preferencias con los datos de la última fecha de la menstruación de la usuaria (04/04/2024), así como la información sobre el ciclo menstrual y su duración en días.

- **Pasos a seguir por la usuaria para dar de alta un registro:**

1.- La usuaria presiona el botón "¿Cuándo fue tu último periodo?" y selecciona la fecha 4 de mayo de 2024 en el calendario. El campo se actualiza con la nueva fecha (Figura 18).

Parámetro E/	Valor
Duración de la menstruación	5 días
Periodicidad de la menstruación	28 días
Fecha de la última menstruación	04/05/2024

Tabla con los nuevos registros: la duración, la periodicidad y la fecha de la regla actual.

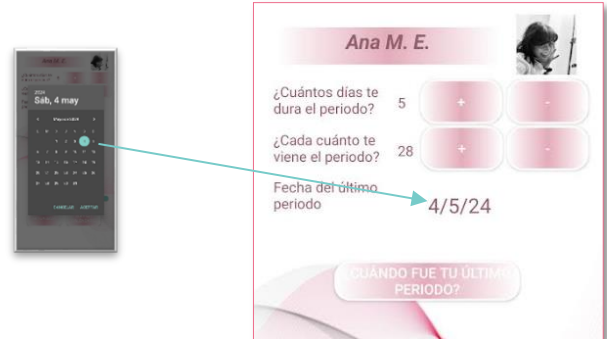


Figura 18: Cuadro de mandos de la Pantalla Preferencias con los datos actualizados por la usuaria.

2.- La usuaria no modifica la duración de la menstruación, que se mantiene en 5 días.

3.- La usuaria decide cambiar la periodicidad de su menstruación a 28 días, para lo cual presiona el botón ↓ una vez. La tabla se actualiza con el nuevo valor.

4.- Con base en la información ingresada por la usuaria, la aplicación ha calculado lo siguiente (Figura 19):

Fecha de la próxima menstruación: La próxima menstruación de la usuaria debería comenzar aproximadamente el 2 de junio de 2024.

Días fértiles: La usuaria ovulará aproximadamente entre el 11 y el 17 de junio de 2024, siendo el 16 el día con mayor probabilidad para que se produzca la ovulación. Estos son los días en los que la usuaria es más fértil.



Figura 19: Calendario Menstrual que refleja los resultados de los cálculos de las fechas del próximo periodo menstrual, de ovulación y de fertilidad.

Consideraciones: Este es solo un ejemplo de cómo se puede usar el cuadro de control de ciclo menstrual. La usuaria puede modificar los valores de acuerdo con su propio ciclo menstrual. Hay que señalar que la información proporcionada por la aplicación es solo una estimación.

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**.

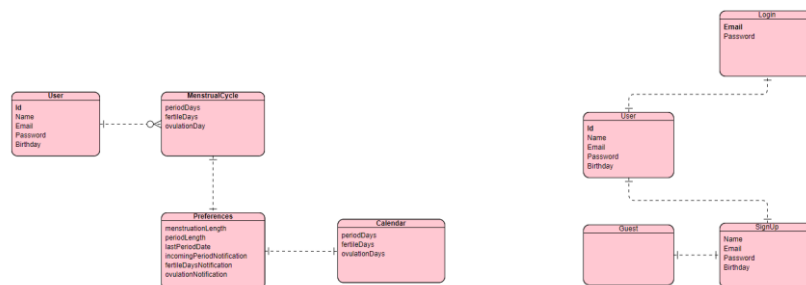
CU7.- Un usuario filtra las notificaciones

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**.

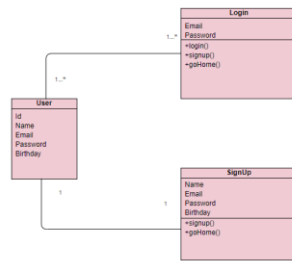
VI.V. Diagrama de E/R:



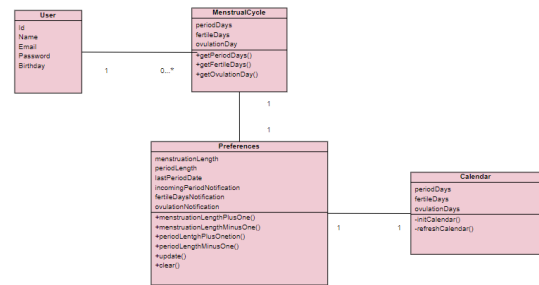
Proceso de Autenticación
y de Alta de Usuario en el sistema

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

VI.VI. Diagrama de clases:

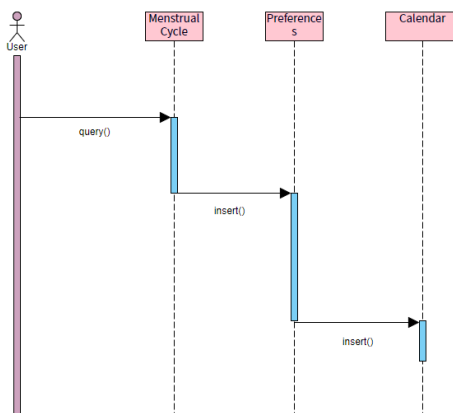


Proceso de Autenticación
y de Alta de Usuario en el sistema



Proceso de creación, actualización y borrado
de registro de

VI.VII. Diagrama de secuencia:



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

VII. Conclusiones

Enfrentarme al diseño y desarrollo de mi primera app móvil ha sido un reto, sobre todo por el hecho de usar una herramienta desconocida como *Android Studio*. Sin embargo, gracias a los conocimientos adquiridos en la asignatura de *Programación Multimedia* y a mi manejo del lenguaje *Java*, logré superar los obstáculos y completar el proyecto.

Si bien la temática del ciclo menstrual no es innovadora, ya que existen diversas aplicaciones similares en el mercado, considero que ha sido una opción adecuada para mi primera incursión en el desarrollo de apps para dispositivos móviles por el nivel de complejidad (no muy elevado) que implica su análisis, diseño y programación.

En cuanto a las áreas de mejora de la app, mencionar que no he logrado el correcto funcionamiento de las notificaciones o alertas, lo cual considero que es una mejora pendiente e importante para completarla. También ha quedado pendiente la implementación de la exportación de datos para compartir con profesionales de la salud, la generación de gráficos o informes sobre el ciclo menstrual de la usuaria, el almacenamiento de los registros de las usuarias y permitirles acceder a su historial menstrual y, por último, la securización de los datos a través de algoritmos de cifrado, de un almacenamiento encriptado y de un control de acceso a personas autorizadas.

A pesar de ello, el desarrollo de esta aplicación ha sido un proceso de aprendizaje continuo que me ha permitido adquirir valiosas habilidades en el ámbito de la programación móvil y me sirve de acicate para seguir formándome en otros lenguajes y frameworks.

VIII. Citaciones y bibliografía

© 2024 Clue de Biowink GmbH, t. l. (s.f.). <https://helloclue.com/es>.
© 2024 Flo Health Inc., F. H. (s.f.). <https://flo.health/es>.
© 2024 MongoDB, l. (2024). *MongoDB*. Obtenido de <https://www.mongodb.com/>
© 2024 womanlog.com, ©. P. (s.f.). <https://www.womanlog.com/es/apps/womanlog>.
Android, D. (s.f.). <https://developer.android.com/studio/intro?hl=es-419>.
Developers, G. (2024). *Firebase*. Obtenido de <https://firebase.google.com/docs/reference/>
Developers, G. (2024). *Google Identity*. Obtenido de <https://developers.google.com/identity/sign-in/web/sign-in?hl=es-419>
Espeso, A. C. (2024). *Programación multimedia y dispositivos móviles (TSDAM)* . Obtenido de <https://campusccc.instructure.com/courses/3215/modules>
Meta, ©. 2. (2024). *MetaforDevelopers*. Obtenido de <https://developers.facebook.com/docs/facebook-login/for-devices>
StackTipsLab. (s.f.). <https://github.com/StackTipsLab/custom-calendar-view/blob/master/library/src/com/stacktips/view/CustomCalendarView.java>.

IX. Anexos

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