**Resumen Ejecutivo**

**Productos y servicios**  
Fitness App es una aplicación móvil diseñada para ayudar a los usuarios del gimnasio a gestionar y optimizar sus entrenamientos. La aplicación permite:

* Registrar ejercicios, incluyendo repeticiones, pesos y descripciones.
* Crear entrenamientos personalizados mediante una galería de ejercicios.

Consultar un historial detallado de progreso.

* Gestionar información personal a través de un perfil de usuario.

Desarrollada en Android Studio con archivos XML para la interfaz y Java para la lógica, Fitness App ofrece una experiencia intuitiva y funcional.

**Beneficios / Ventajas de la aplicación para los usuarios y entrenadores**  
La aplicación aporta numerosos beneficios, tanto para los usuarios individuales como para el sector fitness:

Para los usuarios:

* Organización y planificación mejorada de entrenamientos.
* Seguimiento del progreso para mantener la motivación.
* Experiencia personalizable adaptada a las necesidades individuales.

Para entrenadores:

* Gestión más eficiente de clientes.
* Comunicación clara mediante rutinas estructuradas.

Para el sector fitness:

* Impulso a la digitalización en gimnasios.
* Mayor fidelización gracias a la satisfacción de los usuarios.

En resumen, Fitness App combina tecnología y deporte para crear una herramienta que mejora el rendimiento, facilita la organización y promueve la motivación. Con una interfaz diseñada para ser accesible y práctica, la aplicación se posiciona como una solución moderna e innovadora para el mundo del fitness.

# 1. Creación de Componentes Visuales en la Aplicación

## **1.1. Componentes y Eventos**

La interfaz de usuario de la aplicación ha sido diseñada utilizando **Android Studio**, en la que se emplearon archivos **XML** para definir los componentes visuales, y **Java** para manejar la lógica y los eventos asociados a estos componentes. Entre los componentes visuales utilizados se encuentran:

* **Botones**: Para realizar acciones como la carga y edición de los ejercicios, la navegación entre pantallas, y el envío de datos a la base de datos, además de botones de registro e inicio de sesión.
* **Campos de entrada de texto**: Utilizados para introducir el nombre del ejercicio, el peso, las repeticiones, y descripciones de los ejercicios además de esto todos los campos de registro e inicio de sesión, como los campos para editar datos.
* **Formularios**: Incluyendo el formulario de registro de ejercicios en que hay que rellenar los campos anteriormente comentados.
* **Imágenes**: Asociadas a cada ejercicio para ilustrar el tipo de actividad.
* **Perfil de usuario**: Una sección donde el usuario puede ver y editar su información personal.

Cada uno de estos componentes está asociado a eventos específicos como, por ejemplo:

* **Botones de navegación**: Al hacer clic en estos, se navega entre diferentes actividades dentro de la app.
* **Botones de envío de datos**: Al hacer clic, los datos de los formularios son validados y enviados a la base de datos.
* **Campos de texto**: Detectan cuando el usuario introduce o edita los valores de los ejercicios.

## **1.2.** Propiedades y Atributos

Cada componente tiene atributos definidos en los archivos XML de la interfaz. A continuación, algunos ejemplos como la interfaz de inicio de sesión:

La actividad presentada corresponde a una pantalla de inicio de sesión (**LoginActivity**), donde se han implementado los siguientes componentes básicos:

**TextView:** Para mostrar etiquetas como "Email", "Password" y el título "INICIO SESIÓN".

**TextInputEditText:** Para que el usuario introduzca su correo electrónico y contraseña.

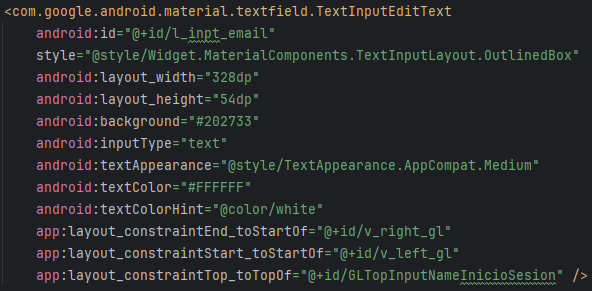
**Button:** Para ejecutar acciones como iniciar sesión o cambiar a la actividad de registro.

**Guidelines:** Utilizadas para organizar y alinear los elementos de forma consistente dentro de un **ConstraintLayout**.

**Botón para iniciar sesión**

****

Este botón llama al método *checkEmptyInputsLogin* al ser pulsado, verificando si los campos de correo y contraseña están vacíos.

**Campo de entrada para el correo electrónico**

Este campo permite al usuario introducir su dirección de correo electrónico, con un fondo oscuro y texto blanco.

En esta actividad, se han configurado propiedades clave para personalizar los componentes visuales.

1. **TextView: Configuración de fuente y color**

**Propiedades destacadas:**

* *android:fontFamily:* Define la tipografía personalizada.
* *android:textSize:* Especifica el tamaño de la fuente en sp.
* *android:textColor:* Configura el color del texto.

1. **Botón: Color de fondo y esquinas redondeadas**

**Propiedades destacadas:**

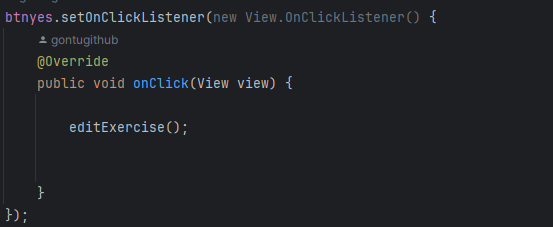
* *android:backgroundTint:* Establece el color del fondo del botón.
* *app:cornerRadius:* Aplica bordes redondeados al botón.

1. **Guidelines: Organización de componentes**

Las Guidelines ayudan a posicionar los elementos en el diseño. En este ejemplo, la guía se encuentra al 10% del ancho de la pantalla.

## **1.3.** Listeners

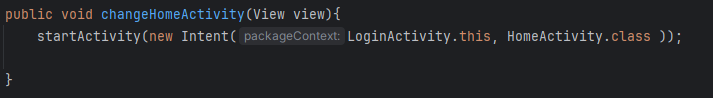
**Listener para confirmar la edición de un ejercicio**

****

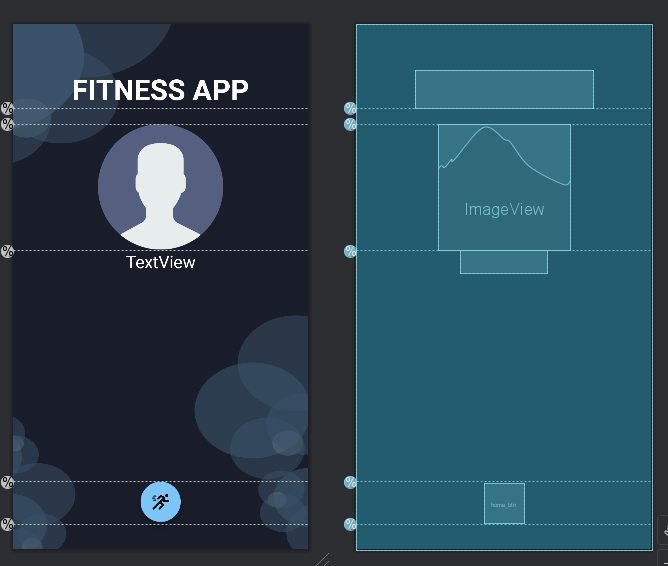
1. *btnYes:* Es el identificador del botón que captura el evento de clic.
2. *setOnClickListener:* Se utiliza para escuchar el evento de clic en el botón.
3. *editExercise():* Método llamado al pulsar el botón, que contiene la lógica necesaria para realizar la edición del ejercicio.

Este Listener asegura que, al pulsar el botón "Yes" (o "Sí"), se confirme la acción de editar un ejercicio en la aplicación. La lógica del método *editExercise* puede incluir la validación de datos, la actualización de la base de datos, y la retroalimentación al usuario, como se muestra a continuación:

## 1.4 Métodos y Eventos

En **Fitness App**, los métodos permiten gestionar diversas acciones dentro de la aplicación, como la navegación entre actividades. Esto es crucial para proporcionar una experiencia de usuario fluida y estructurada.

* *public void changeHomeActivity(View view):* Método público que se ejecuta cuando se vincula a un evento de clic en un botón (por ejemplo, al iniciar sesión).
* *Intent:* Es una clase que permite iniciar nuevas actividades. En este caso, se utiliza para abrir la actividad HomeActivity.
* *startActivity:* Método que lanza la nueva actividad especificada en el Intent.
* *LoginActivity.this:* Hace referencia a la actividad actual desde la que se está llamando el método.
* *HomeActivity.class:* Indica la actividad que se abrirá.



## 1.5 Persistencia

En **Fitness App**, se utiliza una combinación de métodos de persistencia para garantizar la disponibilidad y accesibilidad de los datos, tanto en línea como localmente:

1. **Firebase Realtime Database**: Gestiona los datos principales de la aplicación, como los ejercicios, entrenamientos y perfiles de los usuarios, ofreciendo sincronización en tiempo real.
2. **Base de datos auxiliar en SQLite**: Se utiliza para almacenar localmente los datos de inicio de sesión, permitiendo un acceso rápido y seguro cuando no hay conexión a internet.

#### **1.5.1. Firebase Realtime Database**

Firebase permite almacenar y sincronizar datos en tiempo real entre los usuarios. En **Fitness App**, Firebase se utiliza para guardar los ejercicios y entrenamientos.

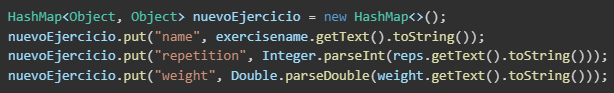
Se crea una instancia de **Firebase Firestore** para poder interactuar con la base de datos.

* Se accede a la colección **users** y se obtiene el documento del usuario correspondiente mediante su *userId*.
* Se usa *Source.SERVER* para asegurarse de obtener los datos actualizados desde el servidor.



Se obtiene la lista de ejercicios almacenada en el documento del usuario.

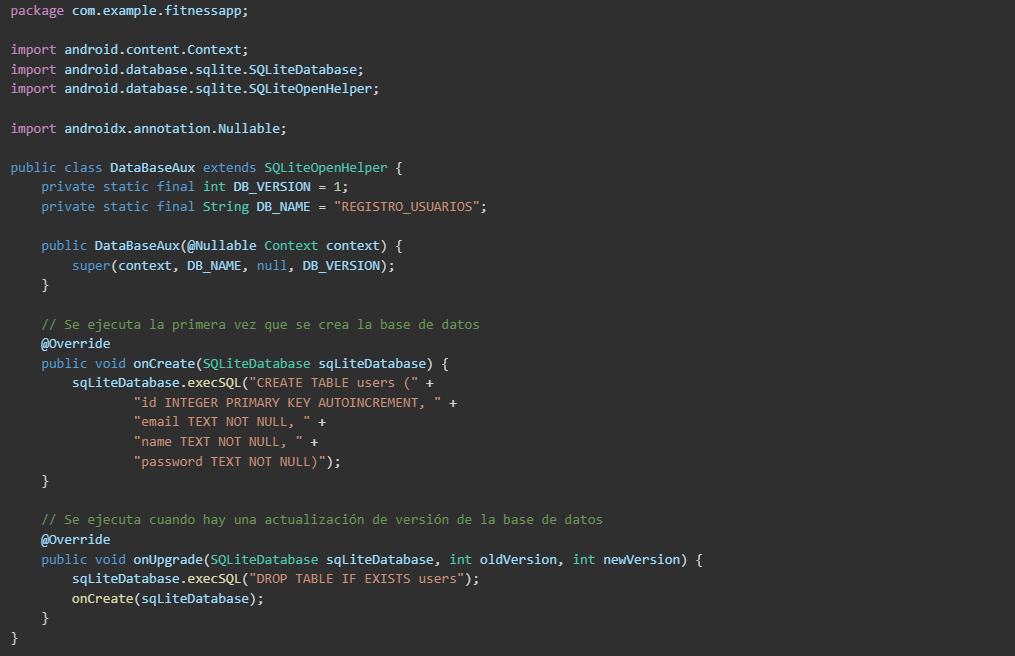
Se verifica si la lista de ejercicios está vacía antes de agregar uno nuevo. Entonces si está vacía agregamos, el Hashmap del nuevo ejercicio

****

#### 1.5.2 Base de Datos Local en SQLite)

En **Fitness App**, además del uso de Firebase, se ha implementado una base de datos local con **SQLite** para almacenar datos de sesión del usuario. Esto permite que la aplicación mantenga la información de inicio de sesión incluso cuando el dispositivo no tiene conexión a Internet.

Este sería el código para poder crear la base de datos local



**Extiende SQLiteOpenHelper**:

Esto permite gestionar la base de datos de manera eficiente.

**Define la estructura de la base de datos**:

* users: Tabla para almacenar los datos de los usuarios.
* Columnas: id (clave primaria), email, name, password.

**Método onCreate**:

* Se ejecuta cuando se crea la base de datos por primera vez.
* Se define la estructura de la tabla users.

**Método onUpgrade**:

* Se usa cuando se actualiza la versión de la base de datos.
* Elimina la tabla existente (DROP TABLE IF EXISTS) y la recrea.

## 1.6 Empaquetado formato APK

El **APK (Android Package)** es el archivo utilizado para instalar aplicaciones en dispositivos **Android**. En **Fitness App**, la generación del APK se realizó con **Android Studio**, permitiendo su instalación y distribución.

#### 1.6.1. Proceso de generación del APK en Android Studio

Para crear el APK, se siguieron estos pasos:

1. **Abrir el proyecto en Android Studio.**
2. Ir a **Build** → *Generate Signed Bundle / APK...*.
3. Seleccionar **APK** y hacer clic en *Next*.
4. **Elegir la clave de firma** (KeyStore) o crear una nueva.
5. Seleccionar la **variante de compilación**:
   * **Debug APK** (para pruebas).
   * **Release APK** (para distribución).
6. Hacer clic en **Finish**.
7. El APK generado se encuentra en:
   * **Modo Release:** app/build/outputs/apk/release/app-release.apk
   * **Modo Debug:** app/build/outputs/apk/debug/app-debug.apk

Una vez generada la APK procederemos a instalarla en nuestro dispositivo móvil Android.

#### 1.6.2. Instalación de la APK

Para hacer la instalación hay 2 formas de hacerlo: ç

#### **Instalación manual**

1. Transferir el archivo APK al móvil.
2. Habilitar **Orígenes desconocidos** en **Ajustes** > **Seguridad**.
3. Abrir el APK y hacer clic en **Instalar**.

#### **Instalación con ADB**

Si el dispositivo está conectado al ordenador con **depuración USB activada**, se puede instalar usando la terminal y ejecutando este comando

 Con el nombre que tenga la APK.

# 2. Usabilidad en Fitness App

La usabilidad en **Fitness App** ha sido diseñada para que los usuarios puedan **navegar e interactuar con facilidad**, asegurando una experiencia fluida e intuitiva. Se han aplicado principios de **accesibilidad, eficiencia y confort visual**, basándose en estándares de usabilidad y ergonomía.

## 2.1. Objetivos de Uso y Estándares de Usabilidad

El diseño de **Fitness App** sigue estos principios clave de usabilidad:

* **Accesibilidad mejorada**: Se ha colocado el menú en la zona inferior, ya que es más fácil de alcanzar con el dedo al sostener el teléfono.
* **Navegación intuitiva**: Cada sección de la app (Inicio, Deportes, Ejercicios) está representada con iconos visuales, permitiendo al usuario identificar rápidamente la función de cada botón.
* **Evitar fatiga visual**: Se ha utilizado un fondo oscuro con tonos azulados, reduciendo la fatiga ocular en el uso prolongado de la aplicación.
* **Tipografía clara y legible**: La fuente elegida facilita la lectura sin causar esfuerzo visual innecesario.

#### **6.2. Aplicación de Normas Internacionales**

##### **ISO / IEC 9126-1 (Calidad del Producto)**

* Se han optimizado los tiempos de carga y la respuesta rápida de la interfaz, asegurando eficiencia y rendimiento en diferentes dispositivos.

##### **ISO / IEC 9241 (Guía de Usabilidad)**

* El menú inferior y la disposición de los elementos permiten que los usuarios interactúen de forma rápida y natural.
* Se ha reducido la sobrecarga de información en cada pantalla, facilitando la navegación.

##### **ISO / IEC 14915 (Ergonomía del Software y Diseño Visual)**

* Uso de iconos representativos en el menú, minimizando la necesidad de texto explicativo.
* Tamaño de botones óptimo para evitar toques accidentales y mejorar la interacción.

## 2.2. Tipos de Usuarios en Fitness App

1. **Usuario Registrado**
   * Puede registrar y gestionar entrenamientos.
   * Accede a su perfil y datos sincronizados en la nube.

## **2.3. Características de Usabilidad Aplicadas en Fitness App**

En **Fitness App**, se han tenido en cuenta diversas características para mejorar la experiencia del usuario, asegurando accesibilidad, claridad y optimización para distintos tipos de dispositivos y usuarios.

#### Capacidades Cognitivas y Perceptivas

* Interfaz clara y minimalista con iconos intuitivos.
* Mensajes de error y confirmación para guiar al usuario.
* Botones diferenciados y bien ubicados para evitar confusión.

Si un usuario intenta guardar un ejercicio sin completar los datos, aparece un mensaje indicando qué falta.

#### Factores Culturales

* Diseño neutro sin referencias culturales específicas.
* Posibilidad de adaptar unidades y formatos en futuras actualizaciones.

Opción de cambiar entre kilogramos y libras según preferencia del usuario.

#### Consideraciones para Discapacidades

* Contraste adecuado para mejorar la legibilidad.
* Botones grandes y bien espaciados para facilitar la interacción.
* Compatible con lectores de pantalla como TalkBack en Android.

Los botones de "Guardar" y "Cancelar" tienen suficiente tamaño para evitar errores al pulsar.