



TÉCNICO SUPERIOR EN DESARROLLO DE
APLICACIONES MULTIPLATAFORMA

Departamento de Informática

PROYECTO



Gacha-Fit

Manual Técnico

Autor: Antonio Crespo Martínez

Curso Académico: 2024-2025

Índice

1 Introducción	2
2 Objetivos	3
3 Tecnologías involucradas	4
4 Proceso de desarrollo	5
4.1. Análisis	5
4.2. Desarrollo (diagrama de secuencias,...)	7
4.3. Pruebas realizadas	9
5 Proceso de Despliegue	10
6 Propuesta de mejora o trabajos futuros	12
7 Bibliografía	13

1 Introducción

Gacha-Fit es una aplicación móvil Android que combina el concepto de fitness con la mecánica de "gacha" (sistema de recompensas aleatorias) para gamificar la experiencia del entrenamiento físico. La aplicación permite a los usuarios realizar ejercicios, registrar su progreso, competir en clasificaciones y coleccionar ejercicios en su "FitDex" (similar a un Pokédex pero para ejercicios).

La aplicación está desarrollada utilizando Java como lenguaje principal y Firebase como backend para la gestión de usuarios, almacenamiento de datos y autenticación. La interfaz de usuario está construida siguiendo las directrices de Material Design, proporcionando una experiencia moderna e intuitiva.

2 Objetivos

Los principales objetivos de Gacha-Fit son:

Gamificar la experiencia fitness: Motivar a los usuarios a realizar ejercicios físicos a través de mecánicas de juego como selección aleatoria de los ejercicios.

Proporcionar una guía de ejercicios: Ofrecer a los usuarios una amplia variedad de ejercicios categorizados por dificultad y tipo.

Registrar el progreso: Permitir a los usuarios hacer un seguimiento de su actividad física y visualizar su progreso a lo largo del tiempo.

Fomentar la competitividad sana: Implementar un sistema de clasificación donde los usuarios pueden comparar su rendimiento con el de otros.

Crear una comunidad de fitness: Establecer un entorno donde los usuarios puedan compartir sus logros y motivarse mutuamente.

Facilitar el aprendizaje de ejercicios: Proporcionar información detallada sobre cada ejercicio, incluyendo instrucciones y beneficios.

3 Tecnologías involucradas

Lenguajes de programación

Java: Lenguaje principal para el desarrollo de la aplicación Android.

XML: Utilizado para la definición de layouts y recursos de la interfaz de usuario.

Frameworks y librerías

Android SDK: Framework principal para el desarrollo de aplicaciones Android.

Material Design Components: Biblioteca de componentes de UI que implementa los principios de Material Design.

Picasso (v2.8): Biblioteca para la carga y caché de imágenes en Android.

CircleImageView (v3.1.0): Componente personalizado para mostrar imágenes en forma circular.

Volley: Biblioteca para la gestión de peticiones HTTP.

Servicios en la nube

Firebase Authentication: Servicio para la gestión de autenticación de usuarios.

Firebase Firestore: Base de datos NoSQL en la nube para almacenar datos de usuarios, ejercicios y progreso.

Firebase Storage: Servicio de almacenamiento para guardar imágenes y otros archivos multimedia.

Firebase Functions: Funciones en la nube para lógica de servidor.

Firebase Analytics: Servicio para analíticas de uso de la aplicación.

Herramientas de desarrollo

Android Studio: IDE principal para el desarrollo de la aplicación.

Gradle: Sistema de automatización de construcción utilizado por Android Studio.

Arquitectura de la aplicación

Arquitectura de múltiples actividades y fragmentos: La aplicación utiliza una combinación de actividades para las principales secciones y fragmentos para las diferentes pantallas dentro de cada sección.

4 Proceso de desarrollo

4.1. Análisis

Diagrama de casos de uso

Los principales casos de uso de la aplicación incluyen:

Registro de usuario

Inicio de sesión

Visualización del perfil

Realización de ejercicios

Consulta del FitDex

Visualización de clasificación

Configuración de preferencias

Cerrar sesión



Título del Proyecto: Gacha-Fit

Diagrama de clases

La aplicación se estructura en varios paquetes:

activity: Contiene las actividades principales como MainActivity, AppCompatActivity, EjercicioActivity y AdminEjerciciosActivity.

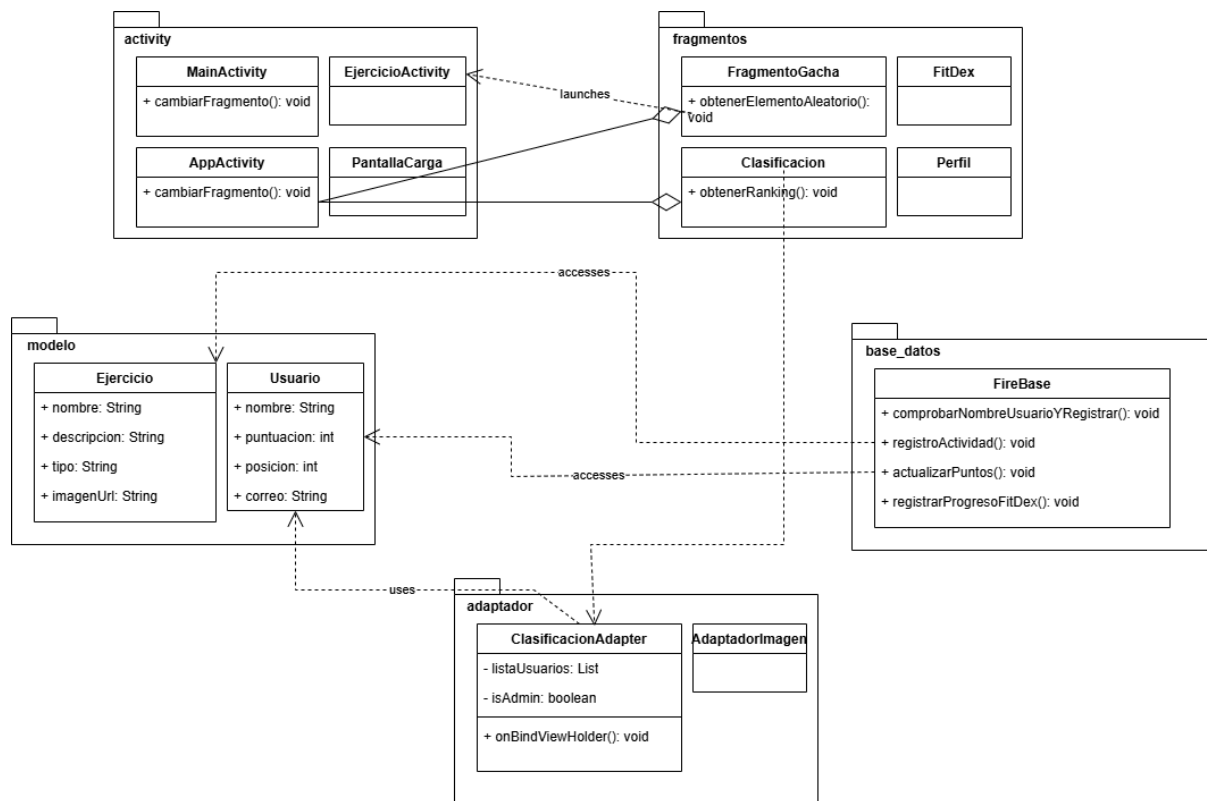
adaptador: Contiene adaptadores para RecyclerViews como ClasificacionAdapter.

base_datos: Contiene clases relacionadas con la conexión a Firebase y gestión de datos, como FireBase y DatosEjercicios.

fragmentos: Contiene todos los fragmentos de la aplicación, como Perfil, FitDex, FragmentoGacha, Clasificacion, entre otros.

modelo: Contiene las clases de modelo de datos.

utiles: Contiene clases de utilidad como Musica.



Modelo de datos

Los principales modelos de datos en Firestore son:

Usuarios: Almacena información de los usuarios (correo, nombre de usuario, contraseña hashada, puntuación).

Ejercicios: Contiene la información de todos los ejercicios disponibles.

FitDex: Registra el progreso de cada usuario en la realización de ejercicios.

RegistroActividad: Almacena un historial de los ejercicios realizados por cada usuario.

4.2. Desarrollo (diagrama de secuencias,...)

Flujo de navegación

La aplicación comienza en PantallaCarga, que verifica la autenticación del usuario. Si el usuario no está autenticado, se redirige a MainActivity que muestra el fragmento de login. Una vez autenticado, se navega a AppActivity, que contiene un menú inferior para navegar entre las diferentes secciones:

Gacha: Pantalla principal donde los usuarios pueden realizar ejercicios aleatorios.

FitDex: Muestra el progreso del usuario en la realización de diferentes ejercicios.

Clasificación: Muestra un ranking de usuarios según su puntuación.

Perfil: Permite al usuario ver y editar su información personal.

Implementación de la autenticación

La autenticación se realiza mediante Firebase Authentication, con un sistema de registro que verifica la disponibilidad del nombre de usuario antes de crear la cuenta. Las contraseñas se almacenan hashadas mediante SHA-256 para mayor seguridad.


```
private String hashSHA256(String input) { 1 usage
    try {
        MessageDigest digest = MessageDigest.getInstance( algorithm: "SHA-256");
        byte[] hashBytes = digest.digest(input.getBytes());
        StringBuilder hexString = new StringBuilder();

        for (byte b : hashBytes) {
            String hex = Integer.toHexString(0xff & b);
            if (hex.length() == 1) hexString.append('0');
            hexString.append(hex);
        }
        return hexString.toString();
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
        return null;
    }
}
```

Sistema de gamificación

El sistema de gamificación incluye:

Un mecanismo de recompensas aleatorias (gacha) para ejercicios.

Un sistema de puntuación basado en la dificultad de los ejercicios completados.

Un "FitDex" donde los usuarios pueden ver qué ejercicios han realizado y cuáles les faltan.

Una tabla de clasificación donde los usuarios pueden comparar su progreso con otros.

Gestión multimedia

La aplicación utiliza Firebase Storage para almacenar imágenes de perfil de usuario y recursos multimedia relacionados con los ejercicios. La biblioteca Picasso se utiliza para cargar y mostrar estas imágenes de manera eficiente.

Sistema de música

La aplicación implementa un sistema de música de fondo que puede ser configurado por el usuario, controlando su volumen o desactivándola completamente.

Autor/es: Antonio Crespo Martínez

4.3. Pruebas realizadas

Pruebas de registro e inicio de sesión

- **Componente:** PantallaLogin y PantallaRegistro
- **Datos de entrada:** Correos electrónicos y contraseñas válidos e inválidos
- **Resultado esperado:** Validación correcta de los datos, registro exitoso, manejo adecuado de errores
- **Herramientas:** Pruebas manuales en dispositivos físicos y emuladores

Pruebas de sistema de gamificación

- **Componente:** FragmentoGacha, FitDex
- **Datos de entrada:** Interacciones de usuario con el sistema de gacha
- **Resultado esperado:** Distribución adecuada de ejercicios según rareza, actualización correcta del FitDex
- **Herramientas:** Firebase Analytics para monitorear interacciones

Pruebas de persistencia de datos

- **Componente:** Firebase, Firestore
- **Datos de entrada:** Operaciones CRUD en datos de usuario y ejercicios
- **Resultado esperado:** Correcta sincronización entre la aplicación y la base de datos
- **Herramientas:** Firebase Firestore Dashboard para verificar integridad de datos

5 Proceso de Despliegue

Software necesario

Para desplegar y ejecutar Gacha-Fit, se requiere el siguiente software:

- **Android Studio:** Versión 2023.1.1 o superior
 - Disponible en: <https://developer.android.com/studio>
- **JDK (Java Development Kit):** Versión 11 o superior
 - Disponible en: <https://www.oracle.com/java/technologies/javase-jdk11-downloads.html>
- **Gradle:** Versión 8.0 o superior (incluido con Android Studio)

Proceso detallado para el despliegue

1. **Clonar el repositorio:**
 - a. git clone <https://github.com/AntonioCrespoMartinez/Gacha-Fit.git>
2. **Abrir el proyecto en Android Studio:**
 - a. Abrir Android Studio
 - b. Seleccionar "Open an existing Android Studio project"
 - c. Navegar hasta la carpeta del proyecto y seleccionarla
3. **Configurar Firebase:**
 - a. Crear un nuevo proyecto en Firebase Console (<https://console.firebase.google.com/>)
 - b. Agregar una aplicación Android con el paquete "com.example.gacha_fit"
 - c. Descargar el archivo google-services.json y colocarlo en la carpeta app/ del proyecto
 - d. Habilitar Authentication, Firestore y Storage en Firebase Console
4. **Configurar Firestore:**
 - a. Crear las colecciones: Usuarios, Ejercicios, FitDex, RegistroActividad
 - b. Configurar reglas de seguridad para proteger los datos
5. **Compilar y ejecutar la aplicación:**
 - a. Conectar un dispositivo Android o configurar un emulador
 - b. Hacer clic en "Run" en Android Studio
 - c. Seleccionar el dispositivo objetivo y confirmar

Configuración para uso en dispositivos

Para instalar la aplicación en un dispositivo Android sin pasar por el desarrollo:

1. Generar un APK en Android Studio:

- a. Build > Build App Bundle(s) / APK(s)
- b. Seleccionar APK y cuando se genere ir a la ubicación del archivo

2. Instalar el APK en el dispositivo:

- a. Transferir el APK al dispositivo
- b. En el dispositivo, navegar hasta el archivo APK y tocarlo para instalarlo
- c. Permitir la instalación de aplicaciones de fuentes desconocidas si es necesario

6 Propuesta de mejora o trabajos futuros

Mejoras funcionales

- Modo multijugador: Permitir que los usuarios compitan directamente entre sí en desafíos de ejercicios en tiempo real.
- Sistema de logros: Crear el sistema de recompensas con medallas, niveles y desafíos diarios/semanales.
- Seguimiento de métricas corporales: Añadir funcionalidad para registrar y visualizar cambios en peso, medidas corporales y otros indicadores de fitness.
- Recuperación de contraseña: Añadir una opción en el registro para recuperar la contraseña mediante correos y códigos.
- Proponer ejercicios: El usuario tendrá una pestaña para enviar un ejercicio con todos sus datos, esto le llegará al administrador y podrá rechazar o aceptar la propuesta.

Mejoras técnicas

- Migración a Kotlin: Convertir progresivamente la base de código de Java a Kotlin para aprovechar sus características modernas y seguridad de tipos.
- Implementación de MVC: Reestructurar la arquitectura de la aplicación siguiendo el patrón Model-View-Controller para mejorar la separación de responsabilidades.
- Tests automatizados: Implementar pruebas unitarias e instrumentadas exhaustivas para garantizar la estabilidad de la aplicación.
- Soporte offline: Mejorar la funcionalidad sin conexión utilizando Room Database para almacenamiento local y sincronización posterior.

7 Bibliografía

Documentación oficial

Android API reference. (s. f.). Android Developers.
<https://developer.android.com/reference>

Firebase. (2024). Firebase Documentation. <https://firebase.google.com/docs>

Material Design. (2024). Material Design Guidelines. <https://material.io/design>

Recursos web

Stack Overflow. (2024). Android Questions.
<https://stackoverflow.com/questions/tagged/android>

Medium. (2024). Android Development Stories.
<https://medium.com/tag/android-development>

GitHub. (2024). Android Sample Projects. <https://github.com/topics/android-sample>