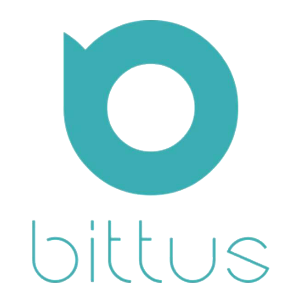
**PROYECTO FITT**

**ORGANIZACIÓN BITTUS**

**MARTES 21 DE MAYO DE 2019**

   
**VERSIÓN 1** 



**MIGUEL ANGEL BELTRAN RODRIGUEZ**

**MIGUEL ANGEL CHAVES CORTES**   
**LAURA JULIANA MORA PÁEZ**   
**JHONNY ALEXANDER PARRA BARRERA**   
**JUAN SEBASTIAN TRIANA PÉREZ**

# Historial de Cambios

|  |  |  |
| --- | --- | --- |
| Fecha | Autor(es) | Descripción del cambio |
| 09/12/2019 | Miguel Beltrán y Jhonny Parra | Primera versión de diagrama de paquetes y descripción. Sección 7.1. |
| 09/12/2019 | Miguel Beltrán y Jhonny Parra | Primera versión de diagrama de despliegue y descripción. Sección 7.2. |
| 12/09/2019 | Laura Mora y Sebastián Triana | Creación BPMN sección 7.3 |
| 13/05/2019 | Miguel Angel Chaves | Primera versión del apartado 8.2. |
| 14/05/2019 | Laura Mora | Descripción BPMN sección 7.3 |
| 16/05/2019 | Miguel Beltrán y Jhonny Parra | Segunda versión de diagrama de paquetes y descripción. Sección 7.1. |
| 16/05/2019 | Miguel Beltrán y Jhonny Parra | Segunda versión de diagrama de despliegue y descripción. Sección 7.2. |
| 16/05/2019 | Miguel Beltrán y Jhonny Parra | Primera versión de diagrama de clases de modelo y controlador. Sección 8.1. |
| 16/05/2019 | Miguel Beltrán y Jhonny Parra | Primera versión de descripción de diagrama de clases de modelo y controlador. |
| 18/05/2019 | Laura Mora y Sebastián Triana | Corrección BPMN sección 7.3 |
| 19/05/2019 | Laura Mora | Primera versión resumen, sección 2 |
| 19/05/2019 | Laura Mora | Primera versión introducción, sección 6. |
| 19/05/2019 | Laura Mora | Segunda versión descripción BPMN sección 7.3 |
| 19/05/2019 | Jhonny Parra y Sebastián Triana | Descripción del JSON de la base de datos documental sección 8.3 |
| 19/05/2019 | Miguel Ángel Chaves | Actualización del apartado 8.2, se agregaron los anexos correspondientes. |
| 20/05/2019 | Sebastián Triana | Finalización descripción de la persistencia del sistema sección 8.3 |
| 20/05/2019 | Miguel Beltrán | Primera y versión final de la sección 8.4 |

# Resumen

En el presente documento se presenta la descripción del diseño de software de la aplicación Fitt, la cual tiene como objetivo ser la herramienta que permita a las personas tener control sobre su actividad física y su progreso a lo largo de la ejecución de las actividades desarrolladas.

El documento se divide en varias secciones donde se dan a conocer los diferentes aspectos de la arquitectura del sistema, utilizando sus diferentes vistas: lógica, física y de procesos; y de esta forma definir la estructura y comportamiento del sistema, al igual que los detalles más importantes del diseño a bajo nivel, los cuales tienen como fin presentar los cambios en el sistema en el tiempo, entre los que se encuentran el almacenamiento de datos y las interfaces con el usuario.

El documento va dirigido a personas que se encuentran interesados en conocer de una forma más detallada la aplicación Fitt desde el punto de vista de su arquitectura, interfaces y procesos, para así conocer su estructura y comportamientos; y que posteriormente le puedan dar el mejor uso a la aplicación y puedan así conseguir el logro de sus objetivos en el desarrollo de la actividad física.

# Tabla de Contenidos

Tabla de contenido

[1 Historial de Cambios 2](#_Toc9289446)

[2 Resumen 3](#_Toc9289447)

[3 Tabla de Contenidos 4](#_Toc9289448)

[4 Lista de Figuras 5](#_Toc9289449)

[5 Lista de Tablas 6](#_Toc9289450)

[6 Introducción 7](#_Toc9289451)

[7 Arquitectura 8](#_Toc9289452)

[7.1 Vista lógica del sistema 8](#_Toc9289453)

[7.2 Vista física del sistema 11](#_Toc9289454)

[7.3 Vista de procesos del sistema 12](#_Toc9289455)

[8 Diseño Detallado 15](#_Toc9289456)

[8.1 Estructura del sistema 15](#_Toc9289457)

[8.2 Comportamiento del sistema 15](#_Toc9289458)

[8.3 Persistencia 16](#_Toc9289459)

[8.4 Interfaz de usuario 19](#_Toc9289460)

[9 Anexos 20](#_Toc9289461)

[10 Referencias 21](#_Toc9289462)

# Lista de Figuras

[Ilustración 1. Diagrama de paquetes Fitt 8](#_Toc9291764)

[Ilustración 2. Diagrama de despliegue Fitt 11](#_Toc9291765)

# Lista de Tablas

# Introducción

En el presente documento se presenta la descripción de diseño de software de la aplicación Fitt, al igual que se tiene como objetivo dar a conocer la definición de la estructura y comportamiento de Fitt, así como los detalles del diseño de bajo nivel de la aplicación.

La información presentada en el presente documento comprende diferentes modelos y diagramas donde se visualiza la arquitectura del sistema, donde unos están enfocados a la vista lógica, vista física o a la vista de los procesos; para así presentar los principales componentes del sistema, al igual que los principales procesos de este. Al igual que se presenta la forma en la que se almacenan los datos y la forma en la que se le muestra al usuario.

La importancia de este documento va enfocada a la orientación detallada que provee para la realización de la aplicación; de forma que proporcione detalles técnicos y de diseño requeridos por el equipo de desarrollo para lograr cumplir con cada uno de los requisitos presentados en documentos anteriores, para así cumplir con las exigencias del cliente y el alcance estipulado.

El sistema de Fitt es una herramienta cuyo fin es permitir a los usuarios tener un control de su actividad física y su progreso a lo largo de la ejecución de las actividades desarrolladas.  Con el fin de lograr esto, el sistema está dividido en 3 módulos: inicio, comunidad y mi perfil.

El módulo inicio cuenta con funcionalidades como: consultar mi proceso, donde se muestran gráficos estilo peso-semanas , también se tiene la  funcionalidad mis logros donde el usuario puede consultar que tanto ha progresado a nivel de logros de la aplicación, se tiene la sección mis rutinas donde se presentan las rutinas asociadas al usuario,  se tienen la sección realizar entrenamiento , donde el usuario podrá ejecutar una entrenamiento, la sección calendario donde  se podrán observar unos entrenamientos programadas para distintos días, la sección entrenador, donde se encuentra el perfil del entrenador asociado al usuario,  iniciar recorrido  y crear rutina donde el usuario podrá crear sus propias rutinas en base a los ejercicios propuestos por el sistema; finalmente se tiene la opción de cerrar entrenamiento.

En el módulo comunidad se presentan funcionalidades para la búsqueda, como la búsqueda de parques, usuarios, rutinas, ejercicios y un mapa con los usuarios que tienen habilitada la opción para mostrar el lugar donde se encuentran entrenando.  Por otro lado, este módulo presenta un chat para comunicarse con el entrenador del usuario.

Finalmente se tiene el módulo de mi perfil donde se muestran algunos datos del usuario como el nombre, una foto, el número de seguidores y de personas que sigue, el nombre del entrenador sin embargo para el caso de que el usuario sea un entrenador no saldrá el nombre del entrenador asociado si no el puntaje en estrellas que tiene.

# Arquitectura

## Vista lógica del sistema

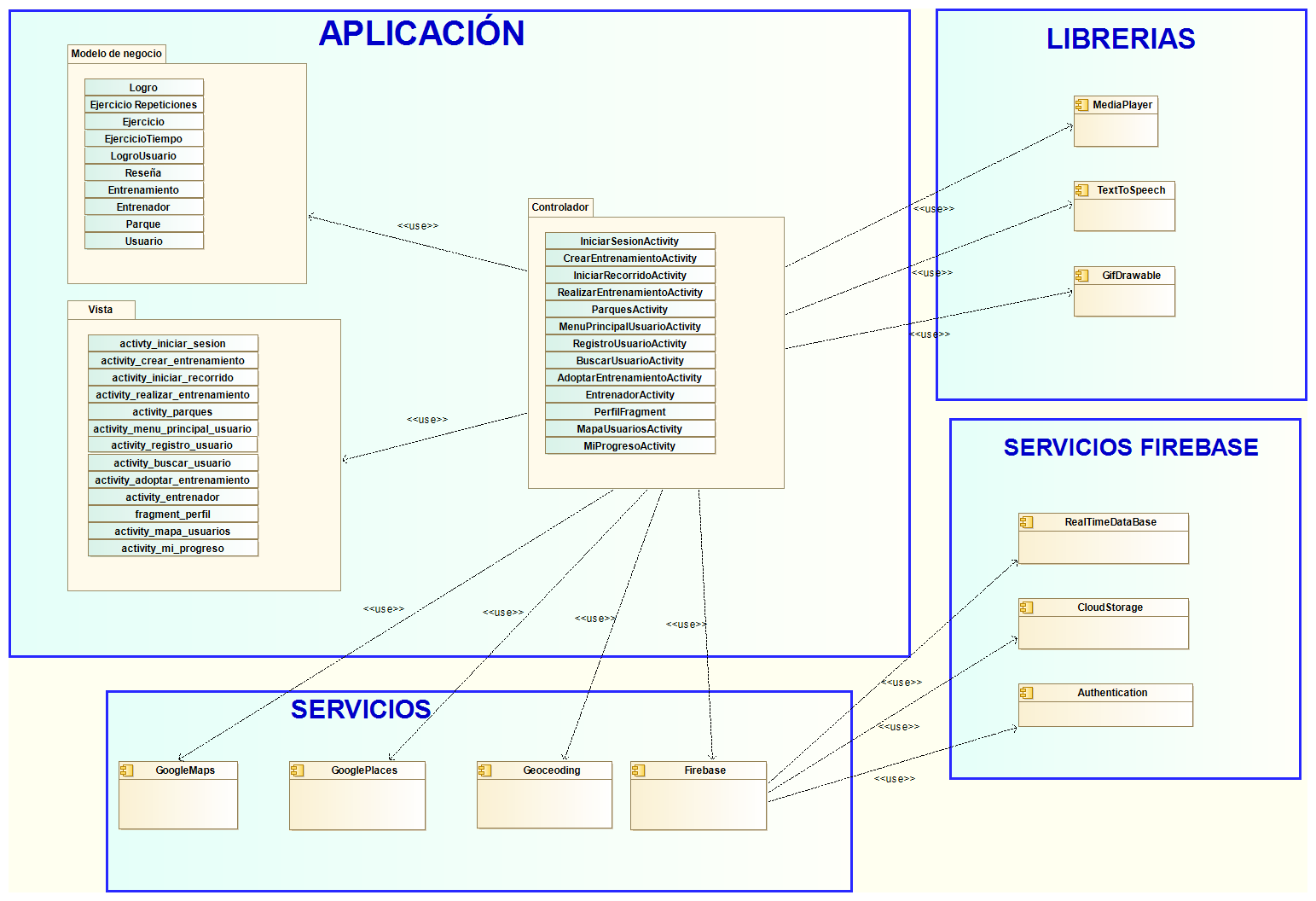


Ilustración 1. Diagrama de paquetes Fitt

El diagrama de paquetes de la aplicación puede ser visto con mayor detalle en el Anexo 1 Diagrama de paquetes.

**Descripción Diagrama de paquetes:**

1. **Aplicación:**

Para modelar la arquitectura de la aplicación se usó el modelo MVC, un patrón de diseño de arquitectura de software para facilitar la separación de los elementos del sistema en tres capas: modelo, vista y controlador:

* 1. **Modelo:**

El paquete de modelo de negocio contiene las clases que guardan los datos de la aplicación, así como la lógica de negocio. En otras palabras, su propósito consiste en almacenar los datos y la lógica de negocio de los objetos del sistema.

* 1. **Vista:**

El paquete de vistas contiene los archivos con extensión “.xml” con la información de las interfaces gráficas del sistema (pantallas, botones, campos de texto, listas, etc). En Android cada una de las vistas está asociada a un controlador (una actividad). El propósito de las clases de este paquete es mostrar los datos de tal forma que puedan ser fácilmente interpretados por el usuario.

* 1. **Controlador:**

El paquete controlador contiene las actividades de la aplicación (en Android, las actividades son clases que actúan como controladores), estas clases se encargan de responder a los eventos originados por los objetos de las vistas y gestionar el flujo de datos entre los objetos del modelo y la capa de visualización. Además, los controladores tienen la responsabilidad de comunicarse con interfaces externas y los servicios de almacenamiento de Firebase, ya sea para descargar o subir la información de los elementos del modelo.

Este patrón de arquitectura fue seleccionado por algunas de sus ventajas que fueron explotadas en el proceso de desarrollo y diseño de la aplicación:

**Desarrollo simultaneo:** mientras algunas personas trabajaban en los elementos de los controladores, otras personas del grupo podían trabajar en las vistas del sistema o en las clases del modelo.

**Facilidad de modificación:** gracias a la separación de responsabilidades, los cambios en el proceso de desarrollo fueron más fáciles de realizar.

**Acoplamiento bajo:** dada la naturaleza del patrón, hay bajo acoplamiento entre modelos, vistas o controladores.

**Alta cohesión:** permite definir de mejor manera las responsabilidades que deben tener cada una de las clases en el proceso de desarrollo.

**Servicios e interfaces externas**

Los servicios e interfaces externas son componentes que no fueron implementados por los desarrolladores del sistema pero que son usados en el funcionamiento del sistema a través de los llamados a sus interfaces:

**Google Maps:** Esta API permite añadir mapas a la aplicación basados en los mapas de Google. La API maneja el acceso a los servidores de mapas de google y se asegura de descargar la información. Adicionalmente, usando está API, se puede realizar operaciones que permiten manipular la información mostrada en los mapas como con añadir marcadores, líneas y polígonos.

**Geocoding**: Esta API permite convertir una dirección o el nombre de una locación en coordenadas geográficas.

**Google Places:** Permite obtener información detallada sobre más de 150 millones de lugares de interés. Esta API retorna información de lugares usando peticiones HTTP.

**Servicios de Firebase:**

**Realtime Database:** Almacena y sincroniza datos de una base de datos NoSQL de la aplicación alojada en la nube. Los datos se sincronizan con todos los clientes en tiempo real y se mantienen disponibles cuando la app no tiene conexión. Los datos son almacenados en formato JSON, se pueden hacer consultas y modificar la información.

**Cloud Storage:** Permite almacenar contenido generado por usuarios, como fotos y vídeos, la información es alojada en la nube. Se pueden hacer descargas de la información, así como subir contenido a la nube.

**Firebase Authentication:** Permite registrar y autenticar a los usuarios a través de un correo electrónico y una contraseña. La información de los usuarios es almacenada en la nube.

**Cloud Mesagging:** Permite enviar mensajes de manera segura y gratuita entre los usuarios de la aplicación.

**Librerías:**

Adicionalmente, vale la pena destacar el uso de algunas librerías especializadas propias del sistema operativo Android que son compatibles con sistemas operativos Android 5.0 Lollipop o superiores y que fueron usadas en el desarrollo de la aplicación:

**MediaPlayer:** Permite reproducir una gran variedad de archivos de audio, video o imágenes dentro de la aplicación.

**TextToSpeech:** Permite hacer síntesis de habla a partir de un texto para reproducción inmediata o para crear un archivo de sonido.

**GifDrawable:** Permite la reproducción de archivos en formato GIF en la aplicación.

## Vista física del sistema

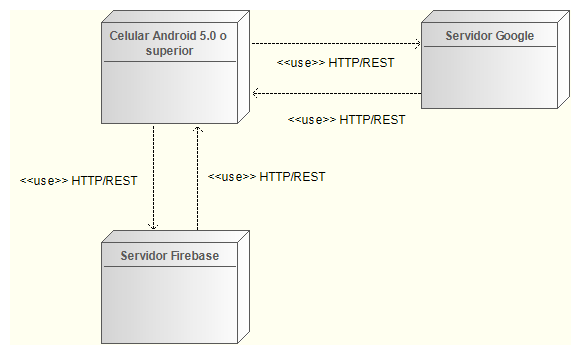


Ilustración 2. Diagrama de despliegue Fitt

**Celular Android 5.0 o superior:**

Es el celular del usuario tiene instalado un sistema operativo Android 5.0 o superior, además, para el correcto funcionamiento del sistema, el celular cuenta con acceso a internet y GPS, en adición de los servicios básicos que ofrece un celular Android. El celular tiene instalada la aplicación y el cliente accede a ella a través del sistema operativo.

**Servidor Google:**

Estos servidores se encargan de almacenar la información y soportar las actividades de procesamiento relacionadas con los servicios de Google utilizados en la aplicación que fueron explicados en la sección 7. Se desconoce la ubicación geográfica, las características de hardware y la forma en la que operan estos servidores. En general, las peticiones realizadas por el celular a estos servidores y las respuestas ofrecidas son transmitidas usando el estilo de arquitectura REST.

**Servidor Firebase:**

Estos servidores almacenan la información de la aplicación, además soportan las tareas de consulta, descarga y modificación de la información. Se desconoce la ubicación geográfica, las características de hardware y la forma en que operan estos servidores. En general, las peticiones realizadas por el celular del cliente y las respuestas ofrecidas por los servidores son transmitidas usando el estilo de arquitectura REST.

## Vista de procesos del sistema

En la presente sección se presentan los principales procesos donde el usuario interactúa con la aplicación Fiit. Para esto se adjunta un BPMN por proceso y se realiza una descripción de cada uno.

Fiit cuenta con 2 actores que interactúan con el sistema, el usuario y el entrenador, sin embargo, para efectos prácticos en los BMPN aparece un pool “Usuario” y durante el proceso se asigna un flujo para el caso donde quien interactúa con el sistema es un usuario o es un entrenador y tengan un flujo diferente.

Los procesos que se identificaron como principales se presentan a continuación:

* **Autenticación:**

Este proceso consta del actor usuario, el cual posee una cuenta ingresa al sistema; para esto el usuario ingresa a la aplicación, posteriormente ingresa correo y contraseña en los respectivos campos y oprime el botón “iniciar sesión”, tras dar oprimir el botón de inicio de sesión el sistema valida que los campos no estén vacíos, para el caso donde los campos estén vacíos se le notifica al usuario que debe llenar todos los campos; cuando se ha validado que los campos no estén vacíos Firebase pasa a validar que la información ingresada coincida con la guardada en la base de datos, para el caso donde la información no coincide se le notifica al usuario y se le pide que vuelva a ingresar correo y contraseña; sin embargo para el caso donde la información coincide con la almacenada el sistema, se realiza una autenticación y se despliega la pantalla de inicio y finaliza el proceso. (Ver Anexo 2 BPMN autenticación.jpeg)

* **Crear entrenamiento:**

Para iniciar este proceso el usuario debe haber iniciado sesión en el sistema, este proceso consta con diferentes fases presentadas a continuación:

* Agregar información básica de entrenamiento: el usuario ingresa a la aplicación, el sistema desplegará la pantalla de inicio del usuario, el usuario oprime el botón “+Entrenamiento” y el sistema desplegará la pantalla de crear entrenamiento, el usuario ingresa los datos del entrenamiento, el usuario oprime el botón “entrenamientos” y el sistema pasa a validar los diferentes campos; para el caso donde los campos son válidos el sistema desplegara la pantalla de agregar ejercicios y para el caso donde no son válidos el sistema notificara al usuario que la información en los campos no es válida y deberá volver a ingresar la información.
* Agregar ejercicios: El usuario oprime el botón “Agregar ejercicio” y el sistema despliega la pantalla buscar ejercicio, el usuario busca el ejercicio y lo selecciona, el sistema define cual tipo de ejercicio ( ejercicio de tiempo, ejercicio de repeticiones, ejercicio de distancia) y despliega el pop-up correspondiente, el usuario ingresa los datos para dicho ejercicio; para el ejercicio de tiempo la cantidad de tiempo que se debe realizar, para el ejercicio de repeticiones deberá ingresar el número de repeticiones que debe hacer, la cantidad de series y el tiempo de descansos, y para los ejercicios de distancia se deberá ingresar a distancia a recorrer; posteriormente el usuario oprime el botón aceptar de dicho ejercicio, el sistema valida los campos del ejercicio, para el caso donde no sean validos se notifica al usuario y se pide que vuelva a ingresar los datos; cuando los campos son correctos se despliega la pantalla buscar ejercicios.
* Agregar descanso: el usuario oprime el botón “Agregar descanso”, el sistema despliega el pop-up de descanso, el usuario ingresa los datos del descanso y oprime el botón “aceptar” el sistema valida los campos del descanso, cuando los campos no son correctos se notifica al usuario y el usuario debe volver a ingresar los datos; para cuando los campos son correctos el sistema despliega la pantalla de agregar ejercicios.
* Guardar entrenamiento: al desplegar la pantalla de agregar ejercicios el usuario oprime el botón “guardar”, el sistema lo guarda y despliega la pantalla de inicio y finaliza el proceso. (Ver Anexo 3 BPMN Crear entrenamiento.png)
* **Registro:**

Este proceso muestra como un usuario se registra en el sistema creando una cuenta. Para iniciar el usuario deberá entrar a la aplicación y el sistema desplegará la pantalla de inicio de sesión, el usuario oprime el botón “registrarse” y elige si se quiere registrar como entrenador o como usuario. Para el caso donde se quiera registrar como entrenador el sistema desplegara la pantalla para registro de entrenador, el entrenador deberá llenar los diferentes campos con su información como correo, contraseña, fecha de nacimiento, peso, altura, genero, titulo, experiencia, y por qué deberían elegirlo como entrenador; posteriormente el entrenador presiona el botón “unirse a Fiit”, el sistema valida que los campos estén llenos y valida que la información sea válida para guardar el entrenador en el sistema y mostrar la pantalla de inicio del entrenador; para el caso donde la información no sea válida o existan campos vacíos, el sistema le notificara al usuario que debe llenar todos los campos o que la información no es válida para que los vuelva a llegar. Para el caso donde se quiera registrar como usuario, el sistema desplegara la pantalla para registro de usuario, el usuario deberá llenar los diferentes campos con su información como correo, contraseña, fecha de nacimiento, peso, altura, genero; posteriormente el usuario presiona el botón “unirse a Fiit”, el sistema valida que los campos estén llenos y valida que la información sea válida para guardar el usuario en el sistema y mostrar la pantalla de inicio del usuario; para el caso donde la información no sea válida o existan campos vacíos, el sistema le notificara al usuario que debe llenar todos los campos o que la información no es válida para que los vuelva a llegar. (Ver Anexo 4 BPMN registro.png)

* **Buscar parque:**

Para iniciar este proceso el usuario debe haber iniciado sesión en el sistema, este proceso consta con diferentes fases presentadas a continuación:

* + Buscar parque: el usuario ingresa a la aplicación, el sistema despliega la pantalla de inicio, el usuario selecciona la pestaña comunidad, el sistema despliega la pantalla comunidad, el usuario oprime el botón “Parques”, el sistema despliega la pantalla parques, despliega el mapa y muestra la ubicación del usuario, Places API genera un URL con la consulta de parques cercanos al usuario, el sistema guarda los datos de la consulta y muestra los parques cercanos al usuario; ahora el usuario tiene dos posibles acciones, ingresar el nombre de un lugar en el buscador para buscarlo u oprimir el marcador de un parque cercano, para el caso donde el usuario decide buscar un lugar, el sistema muestra la ubicación buscada, Places API genera un URL con la consulta de este lugar, el sistema almacena los datos de la consulta y muestra los parques más cercanos a esa ubicación, tras esto el usuario vuelve a tener las dos opciones presentadas anteriormente; para el caso donde el usuario oprima un marcador de un parque cercano el sistema desplegara un pop-up con la información del parque, el usuario oprimirá el pop-up y se desplegara la información detallada del parque, tras esto el usuario tendrá cuatro posibles acciones, finalizar el proceso, oprimir el botón “subir imagen”, oprimir el botón “tomar foto” o oprimir el botón “agregar reseña”; para el caso donde el usuario decida oprimir el botón “subir imagen”, se desplegara la pantalla de selección de imagen, el usuario selecciona una imagen y se pasa a validar la existencia del parque en el sistema para el caso donde no exista el parque se crea un parque y al igual que el caso donde el parque ya existe se pasa a almacenar la imagen en la base de datos y se vuelve a tener las cuatro opciones mencionadas anteriormente; ahora para el caso donde el usuario oprima el botón “tomar foto”, el sistema desplegará la interfaz de la cámara y el usuario tomara la foto, en este punto el proceso se sincroniza con el flujo que se siguió al oprimir el botón “subir imagen” para continuar como se mencionó anteriormente validar la existencia del parque en la base de datos; finalmente para el caso donde el usuario oprima el botón “agregar reseña”, el sistema validara la existencia del parque en la base de datos para el caso donde no exista el parque el sistema creara el parque.
  + Reseñar parque: con un parque ya existente el sistema desplegará un pop-up para ingresar los datos de la reseña, el usuario ingresara los datos de la reseña, el sistema pasara a validar que todos los campos estén llenos, para el caso donde no estén llenos se le dice al usuario que vuelva a llenar los campos, pata cuando los campos están llenos y la información es válida, se almacena la reseña y se vuelve a el punto en la fase “Buscar parque” donde el usuario tiene 4 acciones para realizar. (Ver Anexo 5 BPMN Buscar parque.png)

# Diseño Detallado

## Estructura del sistema

Como fue explicado anteriormente para modelar la arquitectura del sistema, se utilizó un patrón de arquitectura MVC, por lo que se tienen tres paquetes que definen la estructura del sistema: modelo, vista y controlador. Esta sección describe con detalle la estructura de la aplicación Fitt.

En el Anexo 6 Diagrama de clases – Modelo, se muestran únicamente las clases que son propias del modelo de negocio de la aplicación, los constructores, métodos gets y sets fueron omitidos del diagrama para facilitar la lectura de este. En el Anexo 7 Diagrama de clases – Modelo y Controlador se pueden ver las relaciones entre los controladores y las clases del modelo de negocio dentro de la aplicación. Se decidió no incluir en los diagramas la información de las vistas sino únicamente mostrar las relaciones entre los controladores y las clases del modelo de negocio, es importante comprender que cada uno de los controladores (actividades) está asociado a una vista en Android y que, por lo tanto, tiene acceso a la información que estas pueden brindar, por otra parte, las vistas del sistema serán descritas con detalle en la sección 8.4 Interfaz de usuario.

Para una mayor comprensión de las clases, atributos, métodos y asociaciones de cada una de las clases de Fitt se realizó un documento que describe cada una de las clases y sus componentes con detalle, este documento se puede encontrar el Anexo 8 Documentación Diagrama de Clases Fitt.

## Comportamiento del sistema

A continuación, se plasman los procesos más importantes que conforman el núcleo de Fitt respecto a la interacción de los usuarios con el sistema. Para realizar esta descripción se utilizarán diagramas de estados, los cuales son referenciados como anexos a este documento considerando la complejidad del proceso como factor para su descripción detallada.

**Autenticación**

Este proceso permite a los usuarios iniciar sesión en Fitt, Haciendo referencia al CU: *Iniciar Sesión* documentado en el SRS, se muestran las acciones que se requieren para su correcta implementación en el sistema, En el anexo 8.2.1 - Autentificación, se muestra el diagrama con lo anterior.

**Crear Entrenamiento**

Este proceso permite a los usuarios construir su propio entrenamiento a partir de repeticiones ya definidos en la aplicación, esto con el fin de crear nuevo contenido a Fitt y a su vez para adaptar la rutina a cada estilo de vida diferente. Haciendo referencia al CU: *Crear rutina*, *Crear entrenamiento* y *Buscar ejercicio* documentado en el SRS, se muestran las acciones que se requieren para su correcta implementación en el sistema, En el anexo 8.2.2 - Crear nueva rutina, se muestra el diagrama con lo anterior.

**Registro del usuario**

Este proceso permite a nuevos usuarios crear una cuenta dentro de Fitt, esto con el fin de para hacer uso de este. Haciendo referencia al CU: *Registrarse* documentado en el SRS, se muestran las acciones que se requieren para su correcta implementación en el sistema, En el anexo 8.2.3 - Registro del Usuario, se muestra el diagrama con lo anterior.

**Buscar Parque**

Este proceso permite a los usuarios buscar un parque dentro de Fitt, esto con el fin de mostrar lugares aptos para realizar el entrenamiento diario fuera de casa. Haciendo referencia al CU: *Buscar Parque* documentado en el SRS, se muestran las acciones que se requieren para su correcta implementación en el sistema, En el anexo 8.2.4 - Buscar Parque, se muestra el diagrama con lo anterior.

## Persistencia

La información del sistema es almacenada en una base de datos documental NoSQL, por lo que no es necesario, realizar un diagrama de entidad relación.

**Realtime Database:**

En esta sección están almacenada la información de la aplicación en formato JSON de acuerdo a los tipos de datos soportados. El archivo JSON está construido de la siguiente manera:

**ChatList/**

En esta sección se almacena la lista de chats de un usuario, cada lista de chat usa como llave un id único que firebase le otorga, dentro de cada lista se guarda los id de los clientes con los cuales tiene mensajes.

**Chat/**

En esta sección esta almacenada la información básica del chat, cada chat tiene como llave un id único que firebase otorgo y dentro de este tiene asociado un remitente (UID del usuario que manda el mensaje), un receptor (UID del usuario que recibe el mensaje) y un mensaje

**Parques/**

En esta sección está almacenada la información de los parques, cada parqué tiene como llave el nombre del parque y el identificador único que firebase otorga, esto dado que el nombre de la llave no permite usar ciertos caracteres. Adicionalmente tiene la longitud, la latitud, el nombre del parque y una lista de reseñas las cuales contienen un comentario realizado por un usuario, la puntuación que este otorga, la fecha de realización y el usuario que la realiza.

**Usuarios/**

En esta sección esta almacenada la información básica del usuario, tiene como llave el identificador único que firebase otorga. Adicionalmente almacena el correo con que el usuario se registra, la llave del usuario y el nombre del usuario.

**ejercicios/**

En esta sección esta almacenada todos los ejercicios disponibles en la aplicación, cada uno de ellos tiene como llave “ejercicio” y un número que identifica a ese ejercicio en específico. Adicionalmente cada ejercicio contiene su descripción, su dificultad, los músculos que trabaja, el nombre del ejercicio, el elemento que necesita para su realización, la ruta donde se encuentra almacenada el gif en el storage y el tipo de ejercicio.

**entrenamientos/**

En esta sección esta almacenada todos los entrenamientos que un usuario puede crear. Inicialmente crea un campo que usa como llave el UID del usuario asociado, dentro genera por cada entrenamiento creado una llave única que firebase otorga, almacenando la descripción, la dificultad, la duración, el nombre, el número de días de descanso, si el entrenamiento es público o no y dos listas. La primera de ellas contiene los ejercicios del entrenamiento, esta comprende el tiempo de descanso, una lista de ejercicios que usa el entrenamiento, las repeticiones y las series. La segunda lista contiene las reseñas realizadas a esa rutina, esta comprende la calificación, la fecha y el comentario de la reseña.

**entrenamientos\_adoptados/**

En esta sección esta almacenada todos los entrenamientos adoptados por un usuario. Inicialmente crea un campo que usa como llave el UID del usuario asociado, dentro se encontrara los días en que el usuario realizara el entrenamiento, dentro de cada día se genera una llave única que almacenara el día en que realiza el entrenamiento, el entrenamiento y la hora en que la realiza.

**entrenamientos\_públicos/**

En esta ruta se almacenan los entrenamientos públicos de la aplicación, estos entrenamientos, dada su naturaleza, también tienen la información de las reseñas y calificaciones. Inicialmente genera por cada entrenamiento publico una llave única que firebase otorga, almacenando la descripción, la dificultad, la duración, el nombre, el número de días de descanso, si el entrenamiento es público o no y dos listas. La primera de ellas contiene los ejercicios del entrenamiento, esta comprende el tiempo de descanso, una lista de ejercicios que usa el entrenamiento, las repeticiones y las series. La segunda lista contiene las reseñas realizadas a esa rutina, esta comprende la calificación, la fecha y el comentario de la reseña.

**usuarios/**

En esta sección esta almacenada la información específica del usuario, tiene como llave un identificador único que firebase otorga. Adicionalmente almacena la altura, el correo, la dirección donde se encuentra la foto de perfil en el storage, la fecha de nacimiento, el nombre, el peso, la privacidad de la cuenta y el sexo.

**Localizacion\_usuarios/**

En esta sección se almacena la localización de un usuario en el sistema. Inicialmente crea un campo que usa como llave el UID del usuario asociado, dentro se encontrara la latitud, la longitud y el usuario asociado.

**Cloud Storage:**

En esta parte se mantiene toda la información que no puede ser almacenada en la Realtime Database como lo son las fotos de los usuarios, parques y los archivos en formato GIF de los ejercicios. Los archivos son organizados en carpetas de la siguiente manera:

**FotoUsuarios/**

En esta ruta se encuentran almacenadas las fotos de los usuarios. Cada foto está almacenada con un identificador único que asocia la foto con el usuario a quién le pertenece.

**FotoTitulos/**

En esta ruta se encuentran almacenadas las fotos de los títulos de los entrenadores del sistema. Cada foto del título de un entrenador está almacenada con un identificador único que asocia la foto con el entrenador a quién le pertenece el título.

**Imágenes/ejercicios**

En esta ruta están almacenados los GIF correspondientes a cada uno de los ejercicios de la aplicación.

**FotoParques/**

En esta ruta están almacenados las fotos de los parques del sistema, cada parque del sistema está identificado con el nombre del parque seguido de un identificador único que firebase otorga, por lo que las fotos de cada uno de los parques están almacenadas en una subcarpeta en esta ruta con el identificador del parque.

## Interfaz de usuario

Fitt es una aplicación con un “Back-end” o parte lógica brindada por la infraestructura de Firebase es decir, el mayor trabajo para desarrollar la aplicación es en el “Front-end” especialmente en las interfaces gráficas con el usuario.

En esta sección se incluye el Anexo 9 Navegación de Mockups donde se puede apreciar el diseño de las interfaces gráficas además permite navegar en ellas con algunos botones. Es obligatorio aclarar que la navegabilidad de regreso (ir atrás en las pantallas) no está incluida. Esto se puede apreciar más a fondo en el Anexo 10 Diagrama de navegación, donde podemos apreciar un árbol jerárquico de la navegación entre pantallas.

Para el diagrama de navegación es necesario destacar que el punto de partida es la pantalla del usuario. Además, aun siendo un árbol hay algunas conexiones auxiliares para indicar que desde esa pantalla también se puede llegar a una pantalla que ya tiene una pantalla padre.

A continuación, se detallan unos componentes de las interfaces, ya que es necesario brindar información adicional para su correcto funcionamiento. Se debe aclarar que los otros componentes no mencionados son bastante triviales para su implementación, esto es gracias al entorno de desarrollo de Android Studio que brinda herramientas para crear interfaces rápidamente y de manera sencilla.

**Mapas:** En las interfaces gráficas de “mapa usuarios” y “mapa parques” se puede apreciar una imagen con aspecto de mapa. Esta representa un mapa provisto por los servicios de Google por medio de su API. Antes de implementar estos mapas se debe tener una cuenta de Google con medio de pago incluido y verificado, de lo contrario solo permitirá una consulta diaria. Es importante verificar la cuenta, debido a que Google bloqueara todos los servicios incluyendo el de Firebase, esto si la cuenta no se verifica en un corto plazo de tiempo.

**Calendarios:** En las interfaces de “Calendario” se muestra un calendario, este no debe ser un calendario genérico provisto por Android, sino debe ser una implementación completa que permita al usuario interactuar con este, pudiendo seleccionar días y visualizar los días que tiene entrenamientos con un color diferente.

**Gráficas:** En la interfaz de “Progreso” aparecen unas gráficas las cuales representan el progreso del usuario. Estas deben estar con respecto al tiempo como variable independiente y como variable dependiente las calorías, los pasos dados, la distancia recorrida y el tiempo en la aplicación. Se actualizarán cada semana con los datos obtenidos del usuario.

**Gif:** En todas las interfaces donde aparece la imagen de una persona poca abajo se está representando es un gif animado y no una imagen plana. Hay que recordar que todos los ejercicios tienen una animación en formato gif. Para ello se debe utilizar una librería llamada “GifImageDrawable” para poder animar correctamente el gif.

# Anexos

Anexo 1 Diagrama de paquetes.png

Anexo 2 BPMN autenticación.png

Anexo 3 BPMN Crear entrenamiento.png

Anexo 4 BPMN registro.png

Anexo 5 BPMN Buscar parque.png

Anexo 6 Diagrama de clases.png

Anexo 7 Diagrama de clases – Modelo y Controlador.png

Anexo 8 Documentación Diagrama de Clases Fitt.xlsx

Anexo 8.2.1 - Diagrama de Secuencia de Autenticación

Anexo 8.2.2 - Diagrama de Secuencia de Crear Entrenamiento

Anexo 8.2.3 - Diagrama de Secuencia de Registro del Usuario

Anexo 8.2.4 - Diagrama de Secuencia de Buscar Parque

Anexo 9 Navegación de Mockups.pdf

Anexo 10 Diagrama de navegación.png

Anexo 11 Actas entrega 3.xlsx

# Referencias