



WorkOut Share

Joan Josep Sanchez Aznar

Máster universitario de Desarrollo de aplicaciones para dispositivos móviles

Eduard Martin Lineros

07/06/2017



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivad a [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	WorkOut Share – Aplicación para gestionar y compartir tus workouts.
Nombre del autor:	Joan Josep Sanchez Aznar
Nombre del consultor:	Eduard Martin Lineros
Fecha de entrega (mm/aaaa):	06/2017
Titulación:	<i>Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles</i>
Resumen del Trabajo (máximo 250 palabras):	
<p>Este proyecto pretende diseñar e implementar una aplicación en Android que permite al usuario gestionar y compartir workouts.</p> <p>Existe una necesidad constatable de que cada vez más personas se preocupan de su salud, y realizan workouts para mantenerse en forma. El mercado de aplicaciones de gestión de workouts no consta de demasiados competidores pero sí cabe destacar varias aplicaciones que realizan correctamente su función. En este punto la aplicación que se desarrollará en nuestro proyecto opta a ser la más completa de su tipo ya que se va a permitir al usuario nuevas funcionalidades inéditas además de la gestión de sus workouts y estadísticas de progreso.</p> <p>Las funcionalidades claves de nuestro proyecto van a ser: la creación de workouts, la consulta de workouts en un ranking de popularidad, la posibilidad de votar workouts de otros usuarios y por último compartir un workout con amigos. Gracias a estas funcionalidades sociales de consulta, votación y compartición de workouts se va facilitar el uso de la aplicación para los usuarios que no tienen conocimientos o tiempo de crear un workout pero en cambio sí que están dispuestos a consultar los workouts de otros usuarios y elegir uno del ranking de los más populares.</p>	
Abstract (in English, 250 words or less):	
<p>The project aims to design and implement an Android application that will allow the user to manage and share workouts.</p> <p>There is a verifiable need that more and more people care about their health and do workouts to keep fit in. The market for training applications does not have too many competitors but it is worth mentioning several applications which perform their function correctly. After this analysis our application, that will be developed in the project can become the most complete of his kind because it will bring new unpublished functionalities furthermore of the management of the workouts and progress stats data.</p> <p>The key functionalities of the project will be: the creation of workouts, the query of workouts in a popularity ranking, the possibility of voting workouts from other users and lastly sharing a workout with friends. Thanks to this social functionalities of querying, voting and sharing workouts the use of the</p>	

application will be easier for those users which do not have the knowledge or the time to create a workout however they will be willing to check the workouts from other users and choose one from the popularity ranking.

Palabras clave (entre 4 y 8):

Workout Share Gym Planning Android Ranking Progreso Gestión

Índice

1. Introducción	8
1.1 Contexto y justificación del Trabajo	8
1.2 Objetivos del Trabajo	9
1.3 Enfoque y método seguido	9
1.4 Planificación del Trabajo	12
1.5 Breve resumen de productos obtenidos	13
2. Usuarios y contexto de uso	14
2.1 Requisitos funcionales	14
2.2 Diagrama de los casos de uso	15
2.3 Descripción de los casos de uso	15
2.4 Requisitos no funcionales	20
2.4.1 Requisitos look and feel	20
2.4.2 Requisitos de usabilidad y humanidad	20
2.4.3 Requisitos de rendimiento	20
2.4.4 Requisitos de mantenimiento y soporte	21
2.4.5 Requisitos legales	21
3. Diseño conceptual	23
3.1 Diseño de la base de datos	23
3.2 Diseño de las entidades y clases	25
3.3 Diagrama explicativo de la arquitectura del sistema	26
3.4 Escenarios de uso	27
3.4.1 Escenario 1	27
3.4.2 Escenario 2	27
4. Prototipado	29
5. Evaluación	32
5.1 Planificación	32
5.1.1 Fase actual de la planificación	32
5.1.2 Valoración de la planificación	33
5.2 Diseño de la arquitectura	34
5.2.1 Diseño de la base de datos	34
5.2.2 Diseño de las entidades y clases	36
5.2.3 Cambios respecto a la planificación	36
5.3 Prototipo	36
5.3.1 Diseño final de las pantallas	36
5.4 Funcionalidades	39
5.4.1 Registrar una nueva cuenta y Hacer login/logout	39

5.4.2 Crear un WorkOut personalizado	39
5.4.3 Consultar la lista de WorkOuts públicos	39
5.4.4 Consultar la lista de WorkOuts suscritos del usuario	39
5.4.5 Suscribirse/desuscribirse a un WorkOut	39
5.4.6 Mostrar información de un WorkOut	40
5.4.7 Añadir registro de ejercicio y Consultar historial de registros del ejercicio	40
5.4.8 Puntuar un WorkOut	40
5.4.9 Compartir un WorkOut	40
5.4.10 Modificar datos corporales y Consultar historial de datos corporales	40
6. Conclusiones	41
7. Glosario	42
8. Bibliografía	42
9. Anexos	44
9.1 Manual de instalación	44
9.2 Manual de usuario de la aplicación	45

Lista de figuras

<i>Figura 1: Definición de un workout</i>	8
<i>Figura 2: Diagrama de Gantt</i>	12
<i>Figura 3: Diagrama de los casos de uso</i>	15
<i>Figura 4: Diseño teórico de la base de datos</i>	23
<i>Figura 5: Diseño teórico de la base de datos local</i>	24
<i>Figura 6: Diseño teórico de las entidades y clases</i>	25
<i>Figura 7: Diseño teórico arquitectura del sistema</i>	26
<i>Figura 8: Pantallas prototipo</i>	29
<i>Figura 9: Firebase database</i>	34
<i>Figura 10: Firebase BodyData</i>	34
<i>Figura 11: Firebase Users</i>	35
<i>Figura 12: Firebase WorkOuts</i>	35
<i>Figura 13: Clases</i>	36
<i>Figura 14: Diseño final de las pantallas</i>	37

1. Introducción

1.1 Contexto y justificación del Trabajo

Se considera que una persona adulta debería de hacer dos tipos de ejercicio a la semana para mantener su cuerpo sano. Un ejercicio de tipo aeróbico (correr, andar) y un ejercicio de tipo anaeróbico que ejercite la mayoría de músculos (sesión de pesas, flexiones, etc). El conjunto que combina varios de estos ejercicios es lo que se conoce como “workout”.

Definition of *workout* in English:

workout 

NOUN

A session of vigorous physical exercise or training.

'John does daily workouts in the gym'

Figura 1: Definición de un workout

Según fuentes[1] especializadas, uno de cada cuatro españoles está apuntado al gimnasio. Esto puede significar que un 25% de la población del país está o ha estado haciendo un workout en un gimnasio. En este punto puede surgir la intención de analizar, gestionar o cambiar de workout según la necesidad de la persona en cuestión.

Con la facilidad de uso en cualquier lugar que nos aportan los smartphones hoy en día, la idea de una aplicación para gestionar los workouts parece la solución perfecta. El usuario podría hacer uso de ella en el propio gimnasio sin ningún problema o en el lugar que prefiera.

En la Play Store ya existen varias aplicaciones relacionadas con los workouts, las más interesantes podrían ser: *30 Day Fit Challenge Workout, 7 minute Workout, Workout trainer, Fitness & Bodybuilding.*

Estas aplicaciones en principio cubren la necesidad de creación y gestión de un workout, pero todas ellas carecen de una clave por la que nuestro proyecto va a centrarse: Compartir workouts. Hay que asumir que la mayoría de usuarios no debe porqué tener conocimientos de creación de un workout equilibrado, pero los que sí los tengan deben tener la posibilidad de poder compartir sus workouts para que otros usuarios puedan valorarlos y de esta forma sean organizados en un ranking de acceso público.

De esta forma un usuario normal puede acceder a nuestra aplicación sin tener ni la más mínima idea de workouts, consultar el ranking de workouts y elegir uno que haya recibido feedback positivo. Además para finalizar podrá compartir el workout elegido con un amigo de una forma simple.

Con este proyecto se quieren mejorar las opciones de las que dispone un usuario para crear, gestionar, consultar y compartir un workout a través de un smartphone. Se quiere hacer énfasis en el concepto de compartir y valorar workouts ya que no existe en las aplicaciones existentes, y además es un concepto que ayuda a introducir al usuario sin conocimientos en workouts a nuestra aplicación de una forma efectiva.

1.2 Objetivos del Trabajo

El objetivo de este proyecto es diseñar e implementar una aplicación que permita gestionar y compartir ejercicios (workouts).

La primera parte del proyecto consiste en el diseño de las pantallas y menús de la aplicación, mientras que la segunda parte se centra en la implementación del diseño en Android. Esta implementación permitirá al usuario ejecutar varias acciones como: crear un WorkOut personalizado, consultar una lista de WorkOuts, puntuar, compartir un WorkOut y por último permitirá al usuario introducir los datos en su WorkOut para que la aplicación le muestre de forma visual el progreso y estadísticas.

En definitiva los objetivos del proyecto serán los siguientes:

- Diseño de la aplicación.
- Implementación del diseño en Android.

Esta implementación debe ofrecer al usuario las siguientes opciones:

- Permitir al usuario crear un WorkOut personalizado.
- Permitir al usuario consultar una lista de WorkOuts.
- Permitir al usuario puntuar un WorkOut.
- Permitir al usuario introducir los datos en su WorkOut y mostrarle de forma visual su progreso y estadísticas.
- Permitir al usuario compartir un WorkOut.

1.3 Enfoque y método seguido

Lo primero que debemos definir es qué metodología de desarrollo nos interesa más para nuestro proyecto. Vamos a analizar las siguientes metodologías:

Modelo en cascada[2]: El desarrollo se interpreta como una serie de escalones descendentes (cascada de agua) a través de las distintas fases (Análisis, Diseño, Desarrollo, Pruebas, Integración, Mantenimiento) Se hace énfasis en la planificación, los tiempos, fechas objetivo, presupuestos y en la implantación del sistema completo al mismo tiempo

Prototipos[3]: Intenta reducir el riesgo inherente al proyecto dividiendo el proyecto en partes más pequeñas. Se van realizando maquetas a menor escala siguiendo una política de modificaciones hasta culminar los requerimientos de los usuarios.

Incremental[4]: Combinación de metodologías iterativas y lineales con el objetivo primario de reducir los riesgos del proyecto, los proyectos se dividen en partes más pequeñas, de esta manera se facilitan los cambios durante el proceso de desarrollo. Se realizan una serie de mini cascadas, donde todas las fases de desarrollo en cascada se completan para una pequeña parte del sistema, antes de abordar la siguiente parte.

Espiral[5]: Básicamente consiste en una serie de ciclos que se repiten de forma espiral, comenzando desde el centro. Se suele interpretar como que dentro de cada ciclo de la espiral se sigue un modelo cascada pero no necesariamente debe ser así. Este sistema está indicado en grandes proyectos.

Rapid Application Development (RAD)[6]: Comprende el desarrollo iterativo, la construcción de prototipos y el uso de herramientas CASE. Aporta la velocidad de desarrollo, principalmente por el uso de las herramientas CASE. El control del proyecto da prioridad a las fases de desarrollo y define “deadlines”.

En nuestro proyecto vamos a crear un producto nuevo desde cero definiendo unos requerimientos en principio estables, pero el proyecto debe estar preparado para adaptarse a posibles cambios o problemas debido a lo ajustado del calendario.

Por lo expuesto anteriormente se ha decidido optar por una metodología ágil (incremental, iterativa, evolutiva) ya que por la naturaleza del proyecto nos interesa fraccionar las funcionalidades en pequeñas tareas, obtener rapidez en el desarrollo y poder reaccionar y ser flexible a cambios en el proyecto. La metodología ágil con la que trabajaremos en nuestro proyecto será **Scrum**.

Extracto de la wikipedia sobre los beneficios de la metodología Scrum:

Beneficios de Scrum[7]

- **Flexibilidad a cambios.** Gran capacidad de reacción ante los cambiantes requerimientos generados por las necesidades del cliente o la evolución del mercado. El marco de trabajo está

diseñado para adecuarse a las nuevas exigencias que implican proyectos complejos.

- **Reducción del Time to Market.** El cliente puede empezar a utilizar las características más importantes del proyecto antes de que esté completamente terminado.
- **Mayor calidad del software.** El trabajo metódico y la necesidad de obtener una versión de trabajo funcional después de cada iteración, ayuda a la obtención de un software de alta calidad.
- **Mayor productividad.** Se logra, entre otras razones, debido a la eliminación de la burocracia y la motivación del equipo proporcionado por el hecho de que pueden estructurarse de manera autónoma.
- **Maximiza el retorno de la inversión (ROI).** Creación de software solamente con las prestaciones que contribuyen a un mayor valor de negocio gracias a la priorización por retorno de inversión.
- **Predicciones de tiempos.** A través de este marco de trabajo se conoce la velocidad media del equipo por sprint, con lo que es posible estimar de manera fácil cuando se podrá hacer uso de una determinada funcionalidad que todavía está en el Backlog.
- **Reducción de riesgos** El hecho de desarrollar, en primer lugar, las funcionalidades de mayor valor y de saber la velocidad a la que el equipo avanza en el proyecto, permite despejar riesgos efectivamente de manera anticipada.⁴

Por último, el estudiante ejercerá los roles principales que existen en Scrum:

Roles principales

Product Owner	⇒	Estudiante
ScrumMaster	⇒	Estudiante
Equipo Scrum	⇒	Estudiante

Roles auxiliares

Stakeholders	⇒	Usuarios, Profesor
Administrador	⇒	Estudiante

1.4 Planificación del Trabajo

Sprint 1 (PEC1 - Plan de Trabajo)

Tarea 1: Contexto

Tarea 2: Objetivos

Tarea 3: Metodología

Tarea 4: Planificación

Tarea 5: Productos obtenidos

Sprint 2 (PEC2 - Diseño)

Tarea 1: Recogida de requisitos

Tarea 2: Condiciones y contexto de uso

Tarea 3: Escenarios de uso

Tarea 4: Flujos de interacción

Tarea 5: Prototipo del diseño

Sprint 3 (PEC3 - Implementación)

Tarea 1: Creación e integración de Firebase a la aplicación

Tarea 2: Desarrollo de la funcionalidad de creación de un workout

Tarea 3: Desarrollo de la consulta de workouts

Tarea 4: Desarrollo de la funcionalidad de puntuar un workout

Tarea 5: Desarrollo de la funcionalidad de compartir un workout

Tarea 6: Documentación de las decisiones tomadas en la implementación

Sprint 4 (PEC4 - Entrega final)

Tarea 1: Revisión diseño

Tarea 2: Revisión implementación

Tarea 3: Revisión memoria

Diagrama de Gantt:

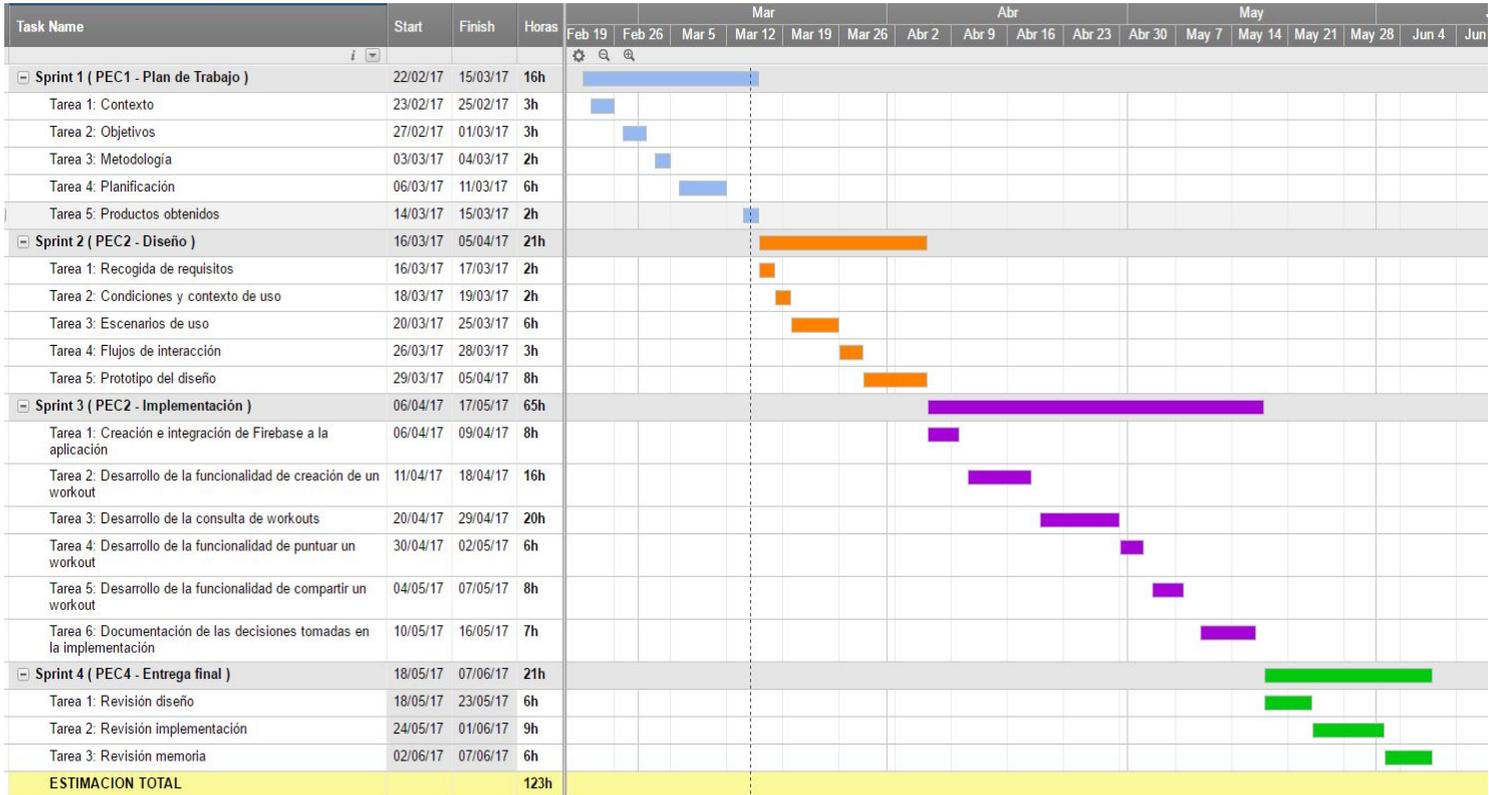


Figura 2: Diagrama de Gantt

1.5 Breve resumen de productos obtenidos

Al concluir el proyecto se obtendrá una aplicación Android que permitirá a los usuarios:

- La creación de workouts personalizados.
- La consulta de workouts ordenados en un ranking de popularidad.
- La recogida de estadísticas y visualización del progreso del usuario.
- La valoración de workouts de otros usuarios.
- Compartir workouts con sus amigos.

2. Usuarios y contexto de uso

El contexto donde opera el proyecto que se quiere desarrollar es el de una aplicación para móvil en la plataforma Android. La aplicación tiene como usuario objetivo a una persona entre 15-65 años que busca cierta automatización en el seguimiento de determinadas actividades físicas para poder evaluar su rendimiento y progreso. Otro tipo de usuario que también está interesado en la aplicación es una persona con pocos conocimientos en los workouts pero que quiere consultar una lista de estos que ya dispongan de un feedback de otros usuarios, de esta manera pueden suscribirse a workouts beneficiosos para ellos.

2.1 Requisitos funcionales

Casos de uso:

- Registrar una nueva cuenta.
- Hacer Login.
- Crear un WorkOut personalizado.
- Consultar la lista de WorkOuts públicos.
- Consultar la lista de WorkOuts suscritos del usuario.
- Suscribirse/Desuscribirse a un WorkOut.
- Mostrar información de un WorkOut.
- Añadir registro de ejercicio.
- Consultar historial de registros del ejercicio.
- Puntuar un WorkOut.
- Compartir un WorkOut.
- Modificar datos corporales.
- Consultar historial de datos corporales.

2.2 Diagrama de los casos de uso



Figura 3: Diagrama de los casos de uso

2.3 Descripción de los casos de uso

ID	C001
Nombre	Registrar una nueva cuenta
Actores principales	Usuario
Precondiciones	1. El actor se encuentra en la pantalla de Login.
Activación	A petición del actor
Escenario principal	<ol style="list-style-type: none"> 1. El actor selecciona la opción de registro en la pantalla. 2. El sistema muestra un formulario de registro. 3. El actor rellena los campos del formulario y finaliza el registro. 4. El sistema registra el nuevo usuario. 5. El caso de uso acaba.
Extensiones	-
Postcondiciones	Nueva cuenta registrada.

ID	C002
Nombre	Hacer Login
Actores principales	Usuario
Precondiciones	1. El actor se encuentra en la pantalla de Login.

Activación	A petición del actor
Escenario principal	<ol style="list-style-type: none"> 1. El actor introduce el usuario y contraseña y hace LogIn. 2. El sistema valida que los datos sean correctos y carga la pantalla principal. 3. El caso de uso acaba.
Extensiones	<p>2a: Los datos introducidos no pasan la validación del sistema.</p> <ol style="list-style-type: none"> 1. El sistema muestra un error en la pantalla de LogIn. 2. El caso de uso acaba.
Postcondiciones	Se ha efectuado el LogIn

ID	C003
Nombre	Crear un WorkOut personalizado
Actores principales	Usuario
Precondiciones	-
Activación	A petición del actor
Escenario principal	<ol style="list-style-type: none"> 1. El actor selecciona crear un WorkOut. 2. El sistema muestra un formulario de creación del WorkOut. 3. El actor rellena los campos del formulario y confirma la creación del WorkOut. 4. El sistema inserta el nuevo WorkOut en la base de datos. 5. El caso de uso acaba.
Extensiones	-
Postcondiciones	Se ha creado un nuevo WorkOut

ID	C004
Nombre	Consultar la lista de WorkOuts públicos
Actores principales	Usuario
Precondiciones	-
Activación	A petición del actor
Escenario principal	<ol style="list-style-type: none"> 1. El actor selecciona consultar el ranking de WorkOuts. 2. El sistema muestra una lista de WorkOuts ordenada por puntuación. 3. El caso de uso acaba.
Extensiones	-
Postcondiciones	Se ha mostrado un ranking de WorkOuts.

ID	C005
Nombre	Consultar la lista de WorkOuts suscritos del usuario
Actores principales	Usuario
Precondiciones	-
Activación	A petición del actor

Escenario principal	<ol style="list-style-type: none"> 1. El actor selecciona consultar los WorkOuts a los que está suscrito. 2. El sistema muestra una pantalla con una lista accesible de los WorkOuts suscritos por el actor. 3. El caso de uso acaba.
Extensiones	-
Postcondiciones	Se han mostrado los WorkOuts suscritos por el actor.

ID	C006
Nombre	Suscribirse/Desuscribirse a un WorkOut
Actores principales	Usuario
Precondiciones	<ol style="list-style-type: none"> 1. Para suscribirse el actor no debe estar suscrito previamente. 2. Para desuscribirse el actor debe estar suscrito previamente.
Activación	A petición del actor
Escenario principal	<ol style="list-style-type: none"> 1. El actor selecciona suscribirse/desuscribirse de un WorkOut. 2. El sistema inserta/elimina el WorkOut en la base de datos local de suscripciones y se informa al actor. 3. El caso de uso acaba.
Extensiones	-
Postcondiciones	Se ha suscrito/desuscrito al actor a un WorkOut

ID	C007
Nombre	Mostrar la información de un WorkOut
Actores principales	Usuario
Precondiciones	-
Activación	A petición del actor
Escenario principal	<ol style="list-style-type: none"> 1. El actor selecciona un WorkOut para obtener su información. 2. El sistema muestra una pantalla con toda la información del WorkOut. 3. El caso de uso acaba.
Extensiones	-
Postcondiciones	Se ha mostrado la información de un WorkOut

ID	C008
Nombre	Añadir registro de ejercicio
Actores principales	Usuario
Precondiciones	<ol style="list-style-type: none"> 1. El actor se encuentra en la pantalla de un ejercicio (dentro de un WorkOut).
Activación	A petición del actor

Escenario principal	<ol style="list-style-type: none"> 1. El actor selecciona añadir un registro de ejercicio. 2. El sistema crea el registro en la base de datos local e informa al actor. 3. El caso de uso acaba.
Extensiones	-
Postcondiciones	Se ha creado un registro de ejercicio

ID	C009
Nombre	Consultar historial de registros de ejercicio
Actores principales	Usuario
Precondiciones	<ol style="list-style-type: none"> 1. El actor se encuentra en la pantalla de un ejercicio (dentro de un WorkOut).
Activación	A petición del actor
Escenario principal	<ol style="list-style-type: none"> 1. El actor selecciona consultar el historial de registros de un ejercicio. 2. El sistema muestra al usuario por pantalla los registros disponibles para el ejercicio. 3. El caso de uso acaba.
Extensiones	-
Postcondiciones	Se ha mostrado al actor el historial de registros de un ejercicio

ID	C010
Nombre	Puntuar un WorkOut
Actores principales	Usuario
Precondiciones	<ol style="list-style-type: none"> 1. El actor se encuentra en la pantalla de información de un WorkOut.
Activación	A petición del actor
Escenario principal	<ol style="list-style-type: none"> 1. El actor selecciona puntuar con un valor numérico el WorkOut. 2. El sistema registra la puntuación para el WorkOut e informa al actor. 3. El caso de uso acaba.
Extensiones	-
Postcondiciones	Se ha puntuado un WorkOut

ID	C011
Nombre	Compartir un WorkOut
Actores principales	Usuario
Precondiciones	<ol style="list-style-type: none"> 1. El actor se encuentra en la pantalla de información de un WorkOut.
Activación	A petición del actor

Escenario principal	<ol style="list-style-type: none"> 1. El actor selecciona compartir el WorkOut. 2. El sistema ofrece al actor una serie de aplicaciones para compartir el WorkOut. 3. El actor comparte el WorkOut seleccionando una aplicación (facebook, whatsapp, etc). 4. El caso de uso acaba.
Extensiones	-
Postcondiciones	Se ha compartido un WorkOut

ID	C012
Nombre	Modificar datos corporales
Actores principales	Usuario
Precondiciones	-
Activación	A petición del actor
Escenario principal	<ol style="list-style-type: none"> 1. El actor selecciona modificar los datos corporales. 2. El sistema muestra un formulario de datos corporales al actor. 3. El actor rellena el formulario de datos corporales. 4. El sistema registra el cambio de los datos corporales en la base de datos local. 5. El caso de uso acaba.
Extensiones	-
Postcondiciones	Se han modificado los datos corporales

ID	C013
Nombre	Consultar historial de datos corporales
Actores principales	Usuario
Precondiciones	-
Activación	A petición del actor
Escenario principal	<ol style="list-style-type: none"> 1. El actor selecciona consultar el historial de datos corporales. 2. El sistema muestra una pantalla con el historial de los datos corporales. 3. El caso de uso acaba.
Extensiones	-
Postcondiciones	Se ha consultado el historial de datos corporales

2.4 Requisitos no funcionales

2.4.1 Requisitos look and feel

ID	R001
Stakeholders	Usuario
Descripción	El sistema tiene que ser atractivo desde el punto de vista de personas jóvenes y adultas.
Justificación	La aceptación del sistema depende en gran medida de la capacidad de ser atractivo.
Fit criterion	Se necesita hacer una encuesta para valorar el cumplimiento del atractivo visual.
Grado satisfacción	3
Grado insatisfacción	5
Fecha	Descripción de la modificación
04/04/2017	Creación del requisito

2.4.2 Requisitos de usabilidad y humanidad

ID	R002
Stakeholders	Usuario
Descripción	La aplicación tiene que ser usable con facilidad.
Justificación	No se debe frustrar al usuario con interfaces complicadas, deben de ser lo más intuitivas posible.
Fit criterion	Más del 80% de usuarios pueden usar el sistema sin ayuda.
Grado satisfacción	2
Grado insatisfacción	4
Fecha	Descripción de la modificación
04/04/2017	Creación del requisito

2.4.3 Requisitos de rendimiento

ID	R003
Stakeholders	Usuario
Descripción	La aplicación tiene que rápida.
Justificación	Se debe evitar que el usuario pierda tiempo innecesario usando la aplicación.
Fit criterion	La aplicación no tarda más de 5 segundos en responder en ningún caso de uso.
Grado satisfacción	5
Grado insatisfacción	4
Fecha	Descripción de la modificación
04/04/2017	Creación del requisito

ID	R004
Stakeholders	Usuario
Descripción	La aplicación debe funcionar las 24 horas del día y los 365 días del año.
Justificación	Los usuarios pueden acceder en cualquier momento a la aplicación.
Fit criterion	La aplicación es accesible siempre.
Grado satisfacción	4
Grado insatisfacción	4
Fecha	Descripción de la modificación
04/04/2017	Creación del requisito

ID	R005
Stakeholders	Usuario
Descripción	La aplicación debe soportar un mínimo de 100 usuarios simultáneos.
Justificación	Hay que dar soporte a un número razonable de usuarios que quieran usar el sistema a la vez.
Fit criterion	Los usuarios pueden interactuar simultáneamente con el sistema sin retardos superiores a 5 segundos.
Grado satisfacción	4
Grado insatisfacción	5
Fecha	Descripción de la modificación
04/04/2017	Creación del requisito

2.4.4 Requisitos de mantenimiento y soporte

ID	R006
Stakeholders	Usuario
Descripción	La aplicación debe de ser fácilmente ampliable con nuevas funcionalidades.
Justificación	La aplicación debe tener facilidad para ampliarse.
Fit criterion	Es posible hacer cambios en la aplicación y no hay ningún sobrecoste en hacerlos respecto a si se hubieran hecho en fase de desarrollo.
Grado satisfacción	4
Grado insatisfacción	5
Fecha	Descripción de la modificación
04/04/2017	Creación del requisito

2.4.5 Requisitos legales

ID	R007
-----------	-------------

Stakeholders	Usuario
Descripción	La aplicación no compromete ningún dato personal de los usuarios.
Justificación	Se debe cumplir la ley vigente sobre la gestión de datos.
Fit criterion	La aplicación cumple con la LOPD.
Grado satisfacción	2
Grado insatisfacción	5
Fecha	Descripción de la modificación
04/04/2017	Creación del requisito

3. Diseño conceptual

3.1 Diseño de la base de datos

En el diseño de nuestra aplicación se han decidido usar dos tipos de bases de datos.

La primera es una base de datos documental online que nos ofrece google con su herramienta firebase. Esta base de datos al ser documental no se puede modelar en UML como una base de datos relacional; por lo tanto se ha elegido el siguiente diagrama para representarla:

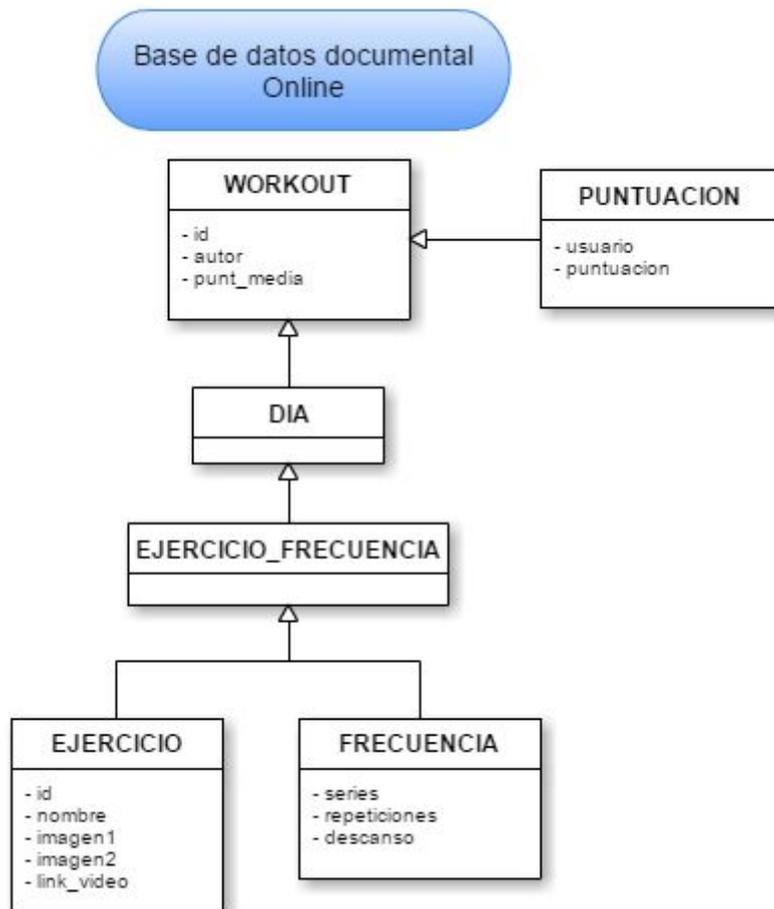


Figura 4: Diseño teórico de la base de datos

La segunda base de datos que usaremos es una local, ya que esto nos permite gestionar las suscripciones de cada usuario y la progresión de los datos corporales y las marcas de los ejercicios. Se puede observar que solo nos guardamos las id's del ejercicio y del WorkOut ya que con este dato consultaremos el resto de información en la base de datos online de firebase.



Figura 5: Diseño teórico de la base de datos local

3.2 Diseño de las entidades y clases

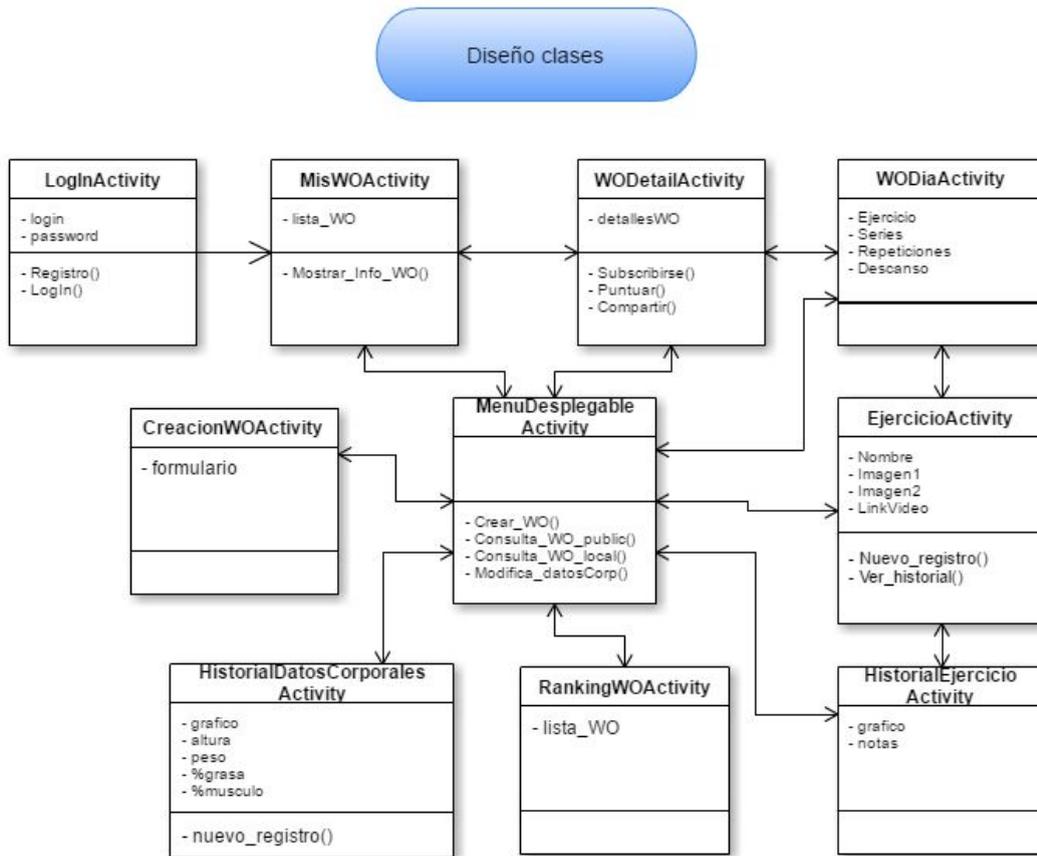


Figura 6: Diseño teórico de las entidades y clases

3.3 Diagrama explicativo de la arquitectura del sistema

En el diagrama se puede observar como en nuestro sistema el usuario usará la aplicación mediante las activities, estas serán las encargadas de acceder a la base de datos online o local y realimentar las vistas de la aplicación con los datos obtenidos para mostrar por pantalla al usuario[8].

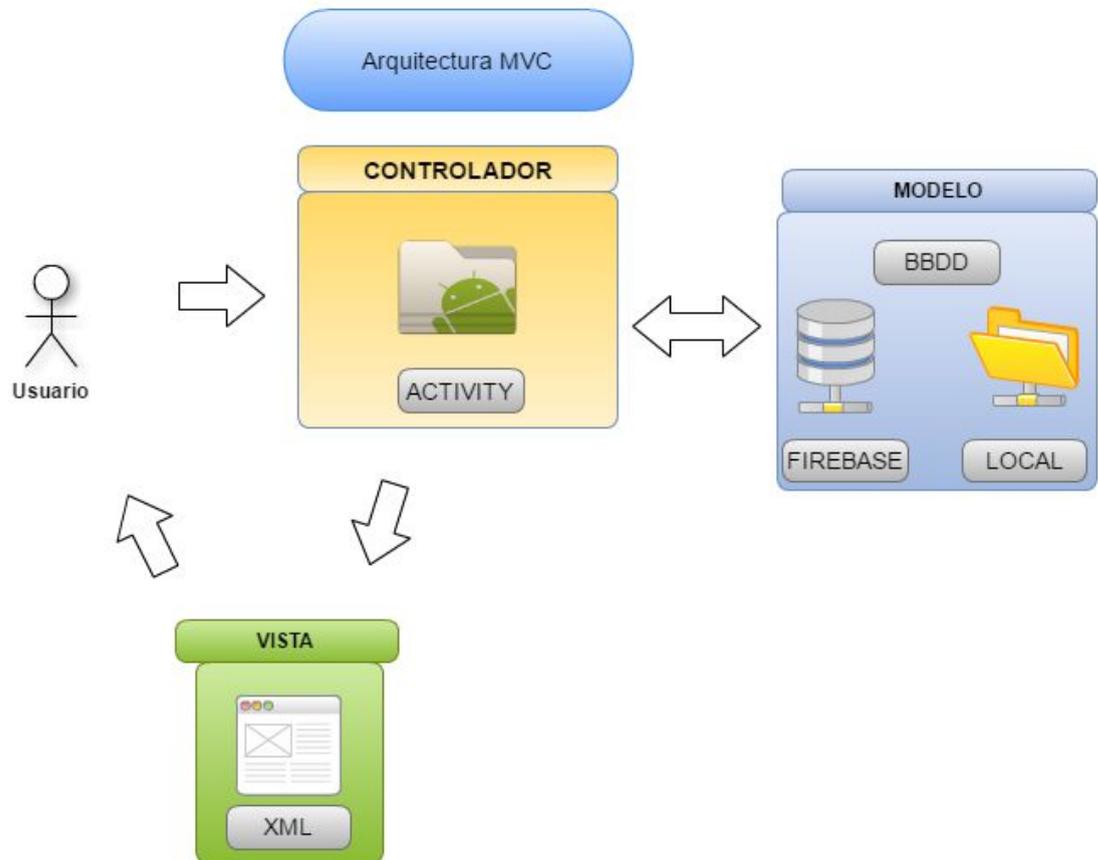


Figura 7: Diseño teórico arquitectura del sistema

3.4 Escenarios de uso

3.4.1 Escenario 1

Nombre: Sergi

Lugar de residencia: Barcelona

Lugar de trabajo: Barcelona

Edad: 24

Sexo: Hombre

S.O. del móvil: Android

Ocupación: Graduado en Sistemas audiovisuales de Telecomunicación, estudiante de máster de desarrollo de aplicaciones para dispositivos móviles y trabajador a jornada completa como programador junior en una consultoría.

Nivel de relación esperado con la app: Frecuente para ir evaluando los progresos en el gimnasio.

Resumen “persona”: Sergi se pasa cada día en el trabajo 9h, es bastante apasionado de la tecnología y el poco tiempo libre del que dispone diariamente lo invierte en sus aficiones tales como los videojuegos de móvil y el gimnasio. Debido al poco tiempo disponible al día piensa que sería ideal disponer de una aplicación para el móvil que le facilite la planificación y evaluación de su rendimiento en el gimnasio.

“Escenario”: Es un lunes a las 18:00h, y como cada lunes Sergi acude a su gimnasio después de trabajar. Mientras se cambia en los vestuarios accede a la app y consulta el workout al que está suscrito para saber que ejercicios le tocan hacer hoy. Sergi va anotando sus marcas en los ejercicios en la app y al final de la sesión se pesa y también lo anota, por último antes de irse a los vestuarios consulta su historial de datos corporales para ver en el historial como ha ganado 2 kilos en los últimos 2 meses y se siente feliz ya que su objetivo de ganar 5 kilos de peso está cada vez más cerca.

3.4.2 Escenario 2

Nombre: Alejandro

Lugar de residencia: Barcelona

Lugar de trabajo: Barcelona

Edad: 25

Sexo: Hombre

S.O. del móvil: Android

Ocupación: Graduado en Ingeniería electrónica, y trabajador a jornada completa como Ingeniero junior en una consultoría.

Nivel de relación esperado con la app: Ocasional, se la acaba de instalar.

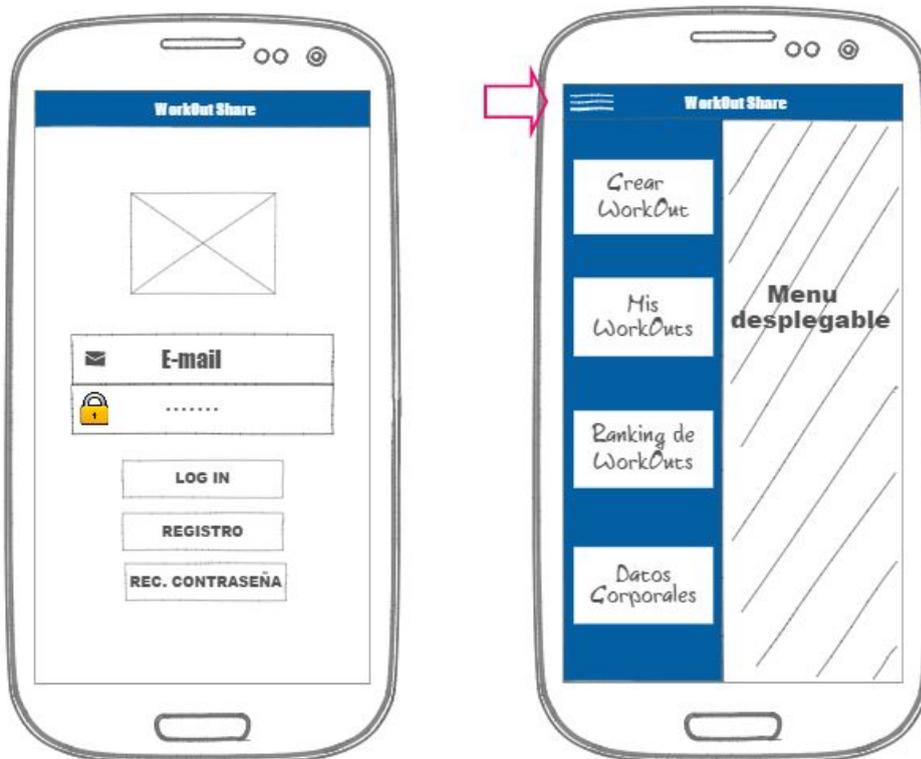
Resumen “persona”: Alejandro se pasa cada día en el trabajo 9h, le apasionan la fotografía y las redes sociales. Dispone de escaso tiempo libre ya que después de la jornada laboral tiene entreno de fútbol. Alejandro quiere mejorar su rendimiento físico para el fútbol y justo acaba de descubrir y bajarse la aplicación.

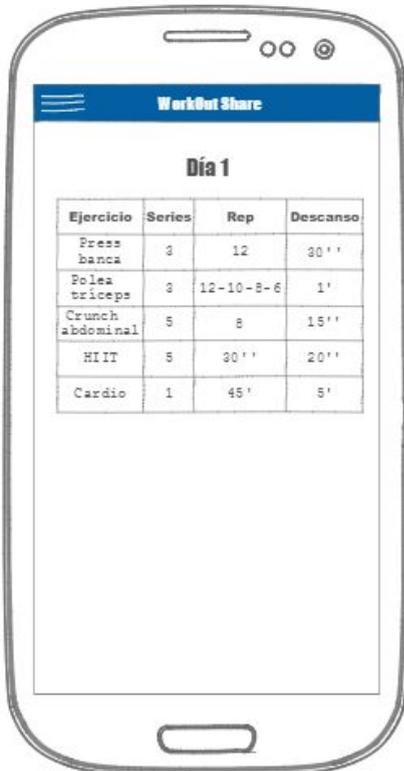
“Scenario”: Es un sábado y Alejandro está interesado en buscar planes de entrenamiento que le ayuden a mejorar su velocidad y resistencia. Se registra en la app y lo primero que hace es consultar el ranking de workouts hasta que encuentra uno que le interesa. Alejandro se lee el workout y observa que el primer ejercicio es correr 45 minutos a ritmo medio, entonces decide que va a bajar a correr a la calle para empezar a trabajar en su objetivo de mejorar su resistencia. Una vez acabado el ejercicio decide introducir su marca para más adelante poder observar si está mejorando.

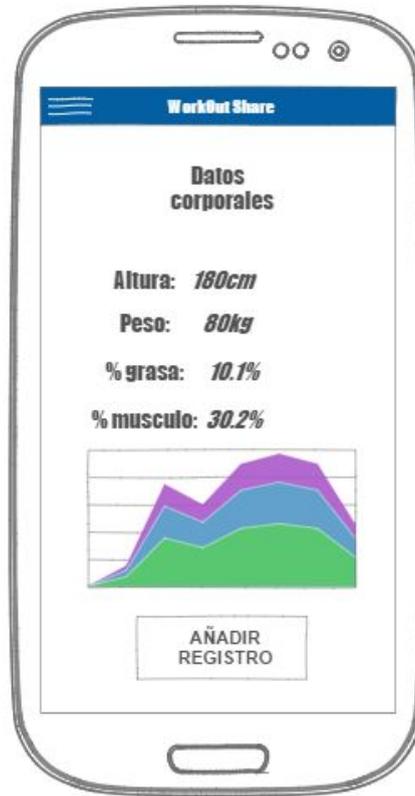
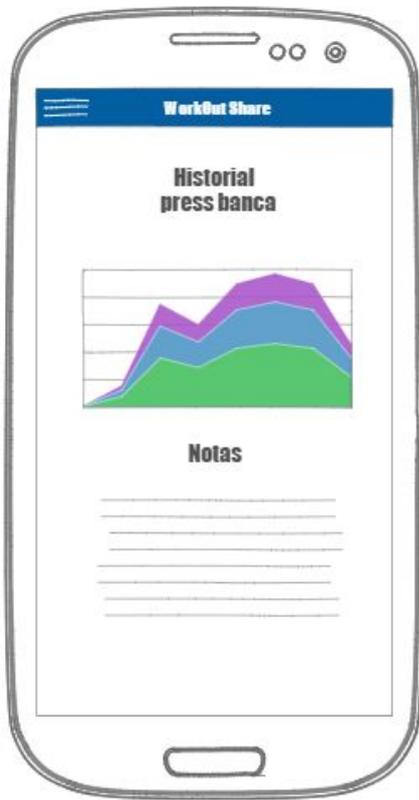
4. Prototipado

Se ha hecho un prototipo de las pantallas más importantes de la aplicación (excepto formularios de registro al ser redundantes). El diseño elegido es el que se procederá a implementar en la fase de desarrollo pero esto no significa que puedan detectarse errores y por lo tanto aplicar algunos cambios[9].

Figura 8: Pantallas prototipo







5. Evaluación

5.1 Planificación

5.1.1 Fase actual de la planificación

Sprint 1 (PEC1 - Plan de Trabajo)

Tarea 1: Contexto

Tarea 2: Objetivos

Tarea 3: Metodología

Tarea 4: Planificación

Tarea 5: Productos obtenidos

Sprint 2 (PEC2 - Diseño)

Tarea 1: Recogida de requisitos

Tarea 2: Condiciones y contexto de uso

Tarea 3: Escenarios de uso

Tarea 4: Flujos de interacción

Tarea 5: Prototipo del diseño

Sprint 3 (PEC3 - Implementación)

Tarea 1: Creación e integración de Firebase a la aplicación

Tarea 2: Desarrollo de la funcionalidad de creación de un workout

Tarea 3: Desarrollo de la consulta de workouts

Tarea 4: Desarrollo de la funcionalidad de puntuar un workout

Tarea 5: Desarrollo de la funcionalidad de compartir un workout

Tarea 6: Documentación de las decisiones tomadas en la implementación

Sprint 4 (PEC4 - Entrega final)

Tarea 1: Revisión diseño

Tarea 2: Revisión implementación

Tarea 3: Revisión memoria

(COMPLETADO)

(MOMENTO ACTUAL)

5.1.2 Valoración de la planificación

En el momento actual estamos situados en el final del Sprint 4 correspondiente a la entrega de la PEC4. Los sprints han sido superados con éxito completando las tareas definidas en la planificación.

Es difícil estimar la precisión de la estimación de horas, pero se podría considerar bastante acertada ya que las horas dedicadas al final han sido ligeramente superiores a las planificadas. Esta diferencia se asume como normal ya que no llega ni a un 10% de desviación total.

5.2 Diseño de la arquitectura

5.2.1 Diseño de la base de datos

Como se ha explicado se ha implementado una arquitectura de base de datos documental en la nube con firebase[10][11].

El esquema principal que se ha diseñado para la aplicación es el siguiente:



Figura 9: Firebase database

Existen 3 nodos principales que son: BodyData, Users y Workouts.

La estructura de los nodos se ha diseñado de esta manera para poder conseguir las relaciones de información necesarias para los usuarios como las suscripciones, registros de datos corporales o ranking de WorkOuts.

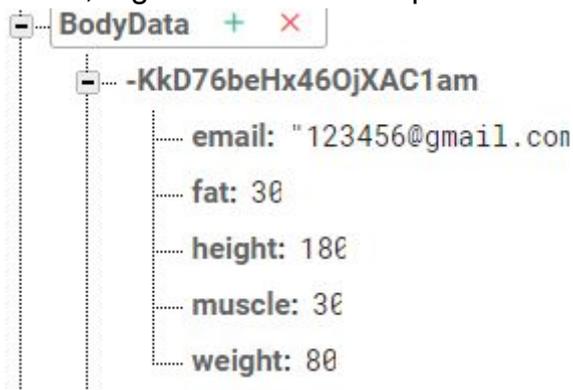


Figura 10: Firebase BodyData

En BodyData se identifica cada registro por una key única y cada registro contiene toda la información necesaria para la aplicación.



Figura 11: Firebase Users

Para el child Users necesitamos el email(login) del usuario y una lista de subkeys que representan la referencia a los WorkOuts suscritos.

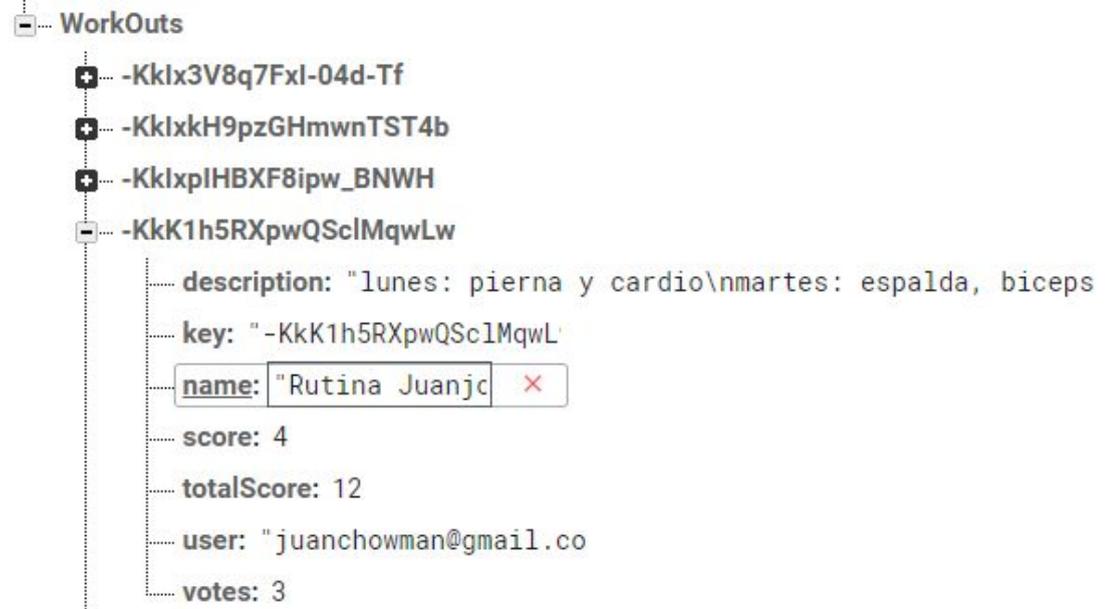


Figura 12: Firebase WorkOuts

Por último para los WorkOuts guardaremos su propia key dentro de su modelo ya que luego será útil cuando se reciban clicks del usuario sobre el recyclerview ya que si no sería imposible identificar el WorkOut clicado en la base de datos.

5.2.2 Diseño de las entidades y clases

En la siguiente imagen podemos observar el diseño usado para las actividades[12]:

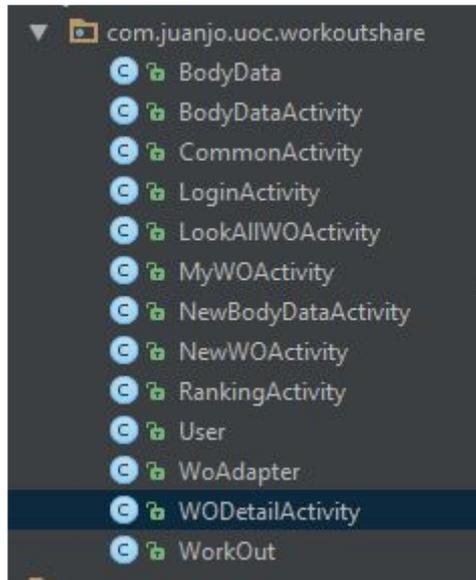


Figura 13: Clases

Las clases WorkOut, User y BodyData son objetos del modelo que se usan para cargar/guardar información en firebase.

Se ha usado una estrategia de polimorfismo al implementar las activities. La activity CommonActivity contiene todos los métodos de conexión a firebase y de esta forma se logra abstraer el resto de actividades y se evita la repetición de código ya que todas las activities heredan de la CommonActivity y por incluyen sus métodos por defecto.

5.2.3 Cambios respecto a la planificación

En la planificación se había mencionado la implementación de una base de datos local. Esta idea se ha descartado por falta de tiempo pero se menciona en las conclusiones como idea de futuro.

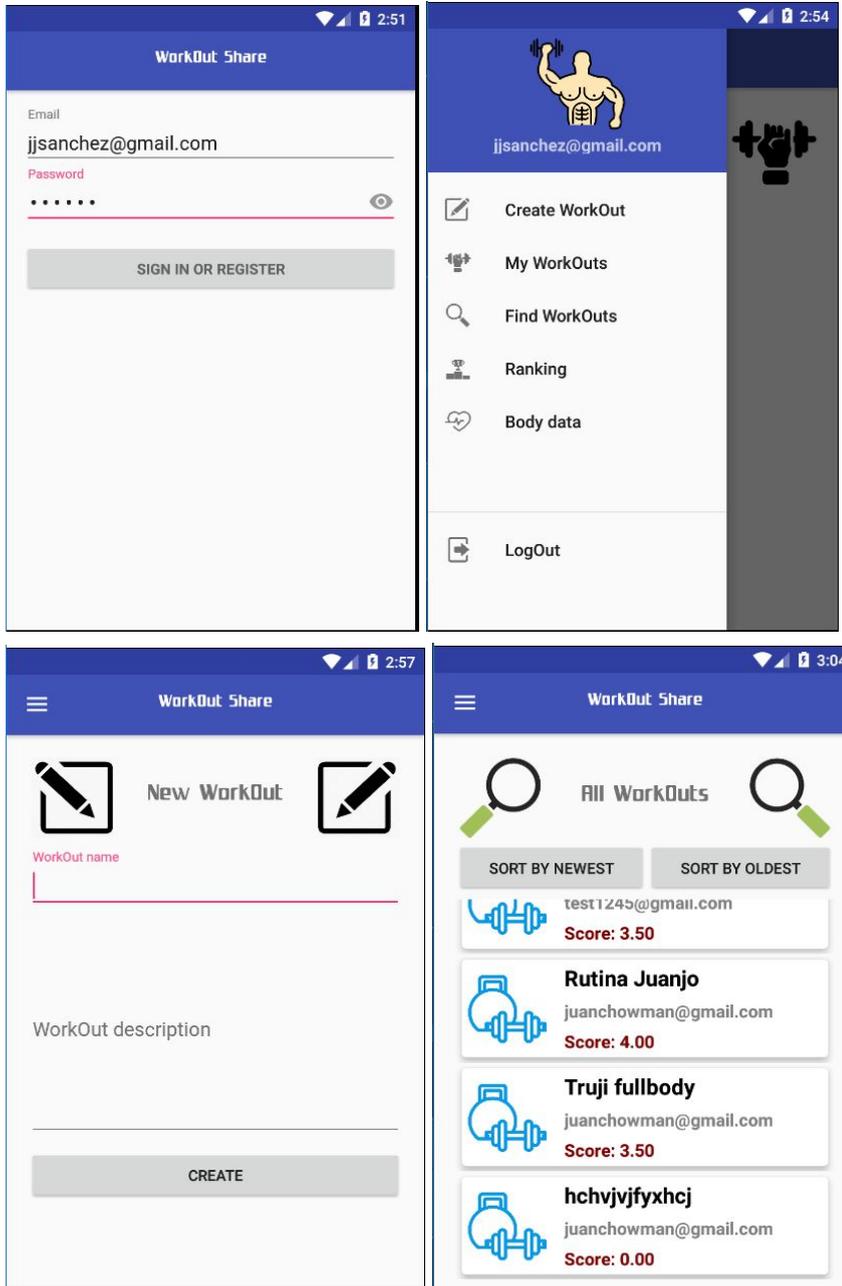
5.3 Prototipo

5.3.1 Diseño final de las pantallas

Como podemos observar se ha intentado mantener la misma estructura que se planteó en el prototipo con un menú desplegable que nos sirve de navegación por todas las pantallas[13] de la aplicación.

Se ha intentado conseguir el mejor resultado estético posible a pesar de que lo más importante y prioritario era conseguir desarrollar las funcionalidades planificadas

Figura 14: Diseño final de las pantallas



WorkOut Share

Ranking

- vhvhh**
test1245@gmail.com
Score: 5.00
- huhuhu**
test1245@gmail.com
Score: 4.50
- Rutina Juanjo**
juanchowman@gmail.com
Score: 4.00
- grgrgrgr**
test1245@gmail.com
Score: 3.50

WorkOut Share

Rutina Juanjo

lunes: pierna y cardio
 martes: espalda, biceps y hiit
 miercoles: descanso
 jueves: hombro, trapecio y cardio
 viernes: pecho, triceps y hiit
 sabado: libre

★ ★ ★ ★ ☆ RATE

SHARE

SUBSCRIBE

WorkOut Share

My WorkOuts

- Rutina Juanjo**
juanchowman@gmail.com
Score: 4.00

WorkOut Share

Bodydata

Height (cm): 175.0
 Weight (kg): 70.0
 Fat %: 18.0
 Muscle %: 30.0

ADD REGISTER

WorkOut Share

New Body data register

Height (cm)

Weight (kg)

Fat %

Muscle %

ADD REGISTER

5.4 Funcionalidades

A continuación se detallan las funcionalidades que se habían planificado y se explica brevemente su implementación:

5.4.1 Registrar una nueva cuenta y Hacer login/logout

Se han implementado en la misma activity el registro de una cuenta y el proceso de login. De esta manera si un usuario sin cuenta intenta hacer login el sistema automáticamente le da de alta una cuenta con el login y password utilizados.

En resumen con la api de firebase se detecta si el usuario está logueado o no. En el caso de estar logueado se cargará la activity principal y se le permitirá navegar y hacer logout. En el caso que no esté logueado se cargará la pantalla de login.

5.4.2 Crear un WorkOut personalizado

Respecto al caso de uso planificado el formulario de creación de un WorkOut ha quedado algo más simplificado por falta de tiempo.

Permite dar un nombre al WorkOut y también contiene un Multiline input text para introducir todo el WorkOut de forma directa. Usando la api de firebase se guardan sus atributos usando el modelo de un objeto de la clase WorkOut generando un nuevo registro en la base de datos de firebase.

5.4.3 Consultar la lista de WorkOuts públicos

En primer lugar se obtiene la lista de WorkOuts mediante la api de firebase y posteriormente gracias a un CustomAdapter se introducen dentro de un componente recyclerview para poder ser mostrados por pantalla con un formato de lista.

También se han implementado dos botones para ordenar por fecha de creación de forma ascendente o descendente. Esto se hace de forma muy sencilla obteniendo la lista de workouts del recyclerview para hacer un reverse de esta y volver a introducirlos en el recyclerview. Por último se hace un invalidate del recyclerview para que se recargue el elemento.

5.4.4 Consultar la lista de WorkOuts suscritos del usuario

La implementación de esta funcionalidad consiste de tres partes principales: obtener las keys identificadoras de los WorkOuts suscritos por el usuario en cuestión, obtener mediante estas keys los objetos WorkOuts referenciados y por ultimo añadir a un recyclerview la lista de WorkOuts resultante para poder visualizarlos en la actividad.

5.4.5 Suscribirse/desuscribirse a un WorkOut

Para implementar la suscripción se obtiene primero la key identificadora del WorkOut para posteriormente añadir en la base de datos la relación entre

usuario y la key. En cambio para desuscribirse de un WorkOut lo que se hará será consultar de la database la relación entre WorkOut keys y usuarios, una vez obtenido el registro correspondiente al usuario en cuestión se procederá a borrar la relación para así completar la suscripción.

5.4.6 Mostrar información de un WorkOut

La implementación de esta funcionalidad consiste en primero obtener la key que identifica al WorkOut del recyclerview. Una vez se obtiene la key, es prácticamente inmediato acceder al WorkOut correspondiente mediante firebase y cargar todos sus atributos en el layout de la actividad de detalle.

5.4.7 Añadir registro de ejercicio y Consultar historial de registros del ejercicio

Estas dos funcionalidades no se han podido implementar a tiempo ya que incrementan la dificultad de gestionar el modelo en la base de datos en la nube con firebase al aumentar demasiado la complejidad de los datos. Aun así en las conclusiones se menciona una futura solución.

5.4.8 Puntuar un WorkOut

Para implementar correctamente esta funcionalidad se ha tenido que añadir al modelo de WorkOut los atributos puntuación total, puntuación y número de votos, ya que era la única forma de poder calcular la media de las votaciones recibidas.

Por lo tanto se ha implementado la funcionalidad de puntuar con una ratingBar de 5 estrellas y gracias a los datos extra que se guardan la puntuación media se recalcula instantáneamente al puntuar.

5.4.9 Compartir un WorkOut

La implementación de esta funcionalidad ha consistido en primero crear un bitmap de la rootView. El segundo paso guarda el bitmap en la SD. Por último creamos un intent para compartir la imagen a través de otra aplicación como whatsapp o facebook.

5.4.10 Modificar datos corporales y Consultar historial de datos corporales

Para esta funcionalidad como necesitamos pintar un gráfico, se ha optado por usar la api <http://www.android-graphview.org/> que implementa las funciones necesarias.

Para crear el gráfico se necesitan pasar los registros a la api, así que nuestro método accede a la base de datos para obtener todos los registros del usuario logeado y se los comunica a la api de graphview. De esta manera se crea un grafico dinamico que se puede ir aumentando sin límite y sirve como un historial visual de los registros.

6. Conclusiones

El diseño y desarrollo de la aplicación WorkOut Share ha proporcionado importantes conocimientos a su autor. Estos conocimientos se resumen en los siguientes:

- **Dominio de firebase:** La api de firebase ofrece herramientas para incorporar una base de datos documental en la nube a tu aplicación. Gracias al uso intensivo que hace WorkOut Share de firebase se considera que se ha dominado esta tecnología y que ahora se cuentan con unos conocimientos que permitirán desarrollos de ideas que antes eran imposibles.
- **Aplicación de técnicas de programación** para la limpieza de código y reutilización de código gracias a la herencia y polimorfismo entre clases.

Se ha logrado implementar casi la totalidad de funcionalidades planificadas. Las funcionalidades no implementadas (Añadir registro de ejercicio y Consultar historial de registros del ejercicio) si bien son importantes no forman parte del núcleo vital de funcionalidades de WorkOut Share.

Se ha seguido la planificación entorno a un 90% de las horas planificadas. La metodología Agile ha servido para la organización y desarrollo de tareas en pequeños sprints que han ayudado a conseguir eficientemente la finalización de éstas.

Las funcionalidades que no se han podido implementar como Añadir registro de ejercicio y Consultar historial de registros del ejercicio se considera que podrían ser parte de un plan de mejoras evolutivas de la aplicación.

Otra línea de trabajo futuro que mejoraría el rendimiento de la aplicación sería una implementación de una base de datos local que se compaginaría con firebase para balancear las llamadas de red a la vez que los tiempos de carga se reducen al no depender 100% de firebase.

7. Glosario

WorkOut: Rutina o conjunto de ejercicios físicos.

Firestore: Api de google que nos ofrece una base de datos documental en el cloud.

Api: Una API es un conjunto de funciones y procedimientos que cumplen una o muchas funciones con el fin de ser utilizadas por otro software. Las siglas API vienen del inglés Application Programming Interface.

Activity: Es el controlador que gestiona la funcionalidad de la vista (pantalla android).

Caso de uso: Descripción de los pasos o actividades que deberán realizarse para llevar a cabo algún proceso.

UML: Unified Modeling Language es un lenguaje de modelado de propósito general en el ámbito de la ingeniería del software que proporciona un estándar para visualizar el diseño de un sistema.

8. Bibliografía

[1]<http://www.estarbien.com/cuerpo-y-mente/2015-10-27/cuidarse-bien/uno-de-cada-cuatro-esp-anoles-esta-apuntado-al-gimnasio/noticia.aspx?idart=945190>

05/04/2017

[2]https://en.wikipedia.org/wiki/Waterfall_model

05/04/2017

[3]https://en.wikipedia.org/wiki/Prototype-based_programming

05/04/2017

[4]https://en.wikipedia.org/wiki/Incremental_funding_methodology

05/04/2017

[5]https://en.wikipedia.org/wiki/Spiral_model

05/04/2017

[6]https://en.wikipedia.org/wiki/Rapid_application_development

05/04/2017

[7][https://en.wikipedia.org/wiki/Scrum_\(software_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development))

17/04/2017

[8]<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>

07/05/2017

[9]<https://ninjamock.com/>

11/05/2017

[10]<https://firebase.google.com/>

15/05/2017

[11]<http://www.json.org/>

15/05/2017

[13]<https://developer.android.com/index.html>

15/05/2017

[14]<https://www.genymotion.com/>

15/05/2017

9. Anexos

9.1 Manual de instalación

La instalación de la aplicación es bastante simple:

1. Ejecutar el archivo .apk en el emulador o dispositivo móvil.
2. Aceptar la instalación de la aplicación.
3. Dar a la aplicación permiso para guardar datos en la memoria.

Una vez instalada ya puedes ejecutar la aplicación sin ningún otro requerimiento, pero hay que tener en cuenta que la aplicación carga de una base de datos en la nube y por lo tanto es recomendable disponer de conexión a internet.

9.2 Manual de usuario de la aplicación

La primera pantalla que vamos a acceder si somos un nuevo usuario es la de login:

Pantalla login: Si el usuario no existe registra el login y si existe te logea. Si por un casual falla el login, usar jjisanchez@gmail.com password: 123456

La siguiente pantalla que accederemos al logear será la de myWorkouts. Pero se puede navegar por la aplicación con el menú desplegable que aparecerá al hacer swipe right o clicar al icono del menú en la esquina superior izquierda.

Pantalla myWorkouts: Se muestran los workouts a los que el usuario está suscrito. Desde aquí puedes entrar en los detalles de un workout y puntuarlo, compartirlo o desuscribirte para que deje de aparecer en myWorkouts.

Pantalla create Workout: Hay que completar el formulario de creación y pulsar el botón create para que la aplicación cree el WorkOut.

Pantalla ranking: Se muestra el top 10 de workouts ordenados por puntuación. Desde aquí puedes entrar en los detalles de un workout y suscribirte/desuscribirte, compartir o puntuar el workout.

Pantalla find Workouts: Se muestran todos los workouts de todos los usuarios y se ofrece la posibilidad de ordenarlos por más reciente o más antiguo. Desde aquí puedes entrar en los detalles de un workout y suscribirte/desuscribirte, compartir o puntuar el workout.

Pantalla body data: Se muestra en un gráfico el historial de marcas de tu peso, % grasa y % músculo. Si se pulsa el botón add register permite añadir una marca nueva que actualizará el gráfico dinámicamente.

Opción share: Si estamos en los detalles de un workout podemos compartirlo a través de una app externa como whatsapp, facebook, twitter o gmail.

Opción LogOut: Permite hacer log out de tu cuenta y carga la pantalla de login.