UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática

TFG del Grado en Ingeniería Informática

-Fatiga+PR

Presentado por Yeray Sardón Ibáñez
en Universidad de Burgos — 1 de julio de 2020
Tutor: José Manual Galán Ordax
Virginia Ahedo García

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática

D. José Manuel Galán, profesor del departamento de Ingeniería de Organización del área de Organización de empresas y Dña. Virginia Ahedo García

Expone:

Que el alumno D. Yeray Sardón Ibáñez, con DNI 71299069N, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado -Fatiga+PR.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección de los que suscriben, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 1 de julio de 2020

Vº. Bº. del Tutor:	Vº. Bº. del co-tutor:
D. José Manuel Galán	D. Virginia Ahedo García

Resumen

-Fatiga+PR es un proyecto que tiene como objetivo monitorear las variables de entrenamiento con el fin de mejorar la fuerza y competitividad en powerlifting.

Para conseguir el objetivo de disminuir la fatiga se llevará el progreso del volumen de entrenamiento que se ha ido realizando diariamente a lo largo de los seis días anteriores. Este progreso se realizará con graficas dado que es el método más visual.

Para conseguir averiguar si se ha conseguido ir mejorando las marcas de competición se calculará las marcas estimadas a partir de tu entrenamiento, con estas aproximaciones podremos ver si la marca mejora indicara que el entrenamiento propuesto funciona, y en caso contrario nos ayudara a saber que se necesitan algunos cambios.

Descriptores

Android, ViewModel, LiveData, Kotlin

Abstract

-Fatiga+PR is a project that aims to monitor training variables in order to improve strength and competitiveness in powerlifting.
To achieve the goal of reducing fatigue, the progress of the training volume that has been carried out daily throughout the previous six days will be taken. This progress will be made with graphics since it is the most visual method.
In order to find out if the competition marks have been improved, the estimated marks will be calculated from your training, with these approximations we will be able to see if the improved mark indicates that the proposed training works, and otherwise it will help us to know that need some changes.

Keywords

Android, ViewModel, LiveData, Kotlin.

Índice general

-Fatiga+PR

Índice de figuras

-Fatiga+PR

Introducción

El fitness es una modalidad que ha ido creciendo mucho en los últimos años, se podría decir que desde la llegada de Arnold Schwarzenegger a la televisión, con películas como Conan o Terminator, ha ido ganando cada vez más adeptos consolidándose como una de las disciplinas deportivas más practicadas en el mundo, sin embargo, este término engloba muchas prácticas deportivas, halterofilia, bodybuilding, strongman, powerlifting o crossfit más recientemente son ejemplos de las diversas prácticas que engloba.

Cada una de las disciplinas tiene su propio sistema para llegar a la excelencia en cada ámbito.

Históricamente uno de los mayores exponentes del powerlifting fue Franco Culumbu, mano derecha de Arnold Schwarzenegger dos veces Míster Olimpia, competición más importante de bodybuilding, y campeón de varios campeonatos de powerlifting. En este tiempo aún no había una estructura de entrenamiento es decir no había un impulsor que impulsara una metodología para el entrenamiento de powerlifting.

Esto cambio con la llegada de Mike Tuchscherer que impulso una metodología basada en RPE (Rate of Percived Exertion), que indica el grado de dificultad de una serie, idea que toma de Anatoly Bondarchuck, lanzador olímpico y entrenador, de su libro Transfer of Training in Sports y de varios alumnos de Anatoly.

El método usado es el de alternar movimientos de competición (Press Banca, Peso Muerto y Sentadilla), con movimientos accesorios en una misma sesión.

Así en una sesión se realiza un movimiento de competición, un movimiento para mejorar técnicamente un punto débil y un movimiento para mejorar un musculo más débil.

Así por día se consigue una mejora en técnica y calidad muscular,

usar los movimientos de competición, (Press Banca, Peso Muerto, Sentadilla), variaciones del movimiento de competición ,con el uso de pausas, pines cadenas u otras modalidades añadidas a estos movimientos de competición, y movimientos accesorios, que son movimientos que se alejan mucho del movimiento de competición sin embargo entrenan el musculo que ayuda a los movimientos de competición, estos son los

-Fatiga+PR

ejercicios que se pueden considerar más clásicos en el gimnasio como poleas extensiones de cuádriceps remos etc.

Mike Tuchscherer crea un sistema que se basa en el RPE para manejar la fatiga a lo largo del tiempo de entrenamiento, tanto diario como semanal o mensual, el objetivo último es estar siempre preparado para realizar tu siguiente entrenamiento sin tener fatiga acumulada del día anterior o la mínima posible, además de introducir elementos para mejorar la fuerza muscular y la técnica todo con objetivo de levantamiento de más peso.

Ajustándome a estos criterios en este trabajo se ha querido realizar una aplicación que se adapte a esta modalidad de entrenamiento realizada por Mike Tuchscherer, además de ajustarse como diario de entrenamiento pudiendo llevar un control del volumen realizado

Objetivos del proyecto

Dividiremos los objetivos en tres categorías:

2.1 Objetivos a cumplir por la aplicación

- Tener un diario en el cual apuntar tu progreso
- Poder volver a el para analizar los posibles fallos en el entrenamiento
- Visualización de en gráficos de las distintas opciones

2.2 Objetivos técnicos

- Uso del lenguaje de programación kotlin y utilización de las características del lenguaje para potenciar la aplicación.
- Uso de las funcionalidades que brinda Android a la hora de manipular y acceder el sistema operativo
- Uso de la tecnología Room para la manipulación de bases de datos SQLite
- Uso de fragmentos para enmascarar varios usos dentro de un mismo UI.
- Uso de barras de navegación para crear un entorno mas amigable y fácil de usar

2.3 Objetivos Personales

- Aumentar el conocimiento en Android y las muchas tecnologías que lo componen
- Facilitar la lectura de las variables de entrenamiento

Conceptos teóricos

3.1 Android

La aplicación será una aplicación Android para monitorear la actividad a lo largo del tiempo.

¿Por qué Android?

La aplicación se realizará en Android, se ha elegido este sistema operativo dado que es el sistema operativo mas usado en el parque de móviles, (1) con mas de un 80% de los dispositivos usados, por lo que para llegar a la mayor cantidad de usuarios posibles la aplicación será realizada en Android.

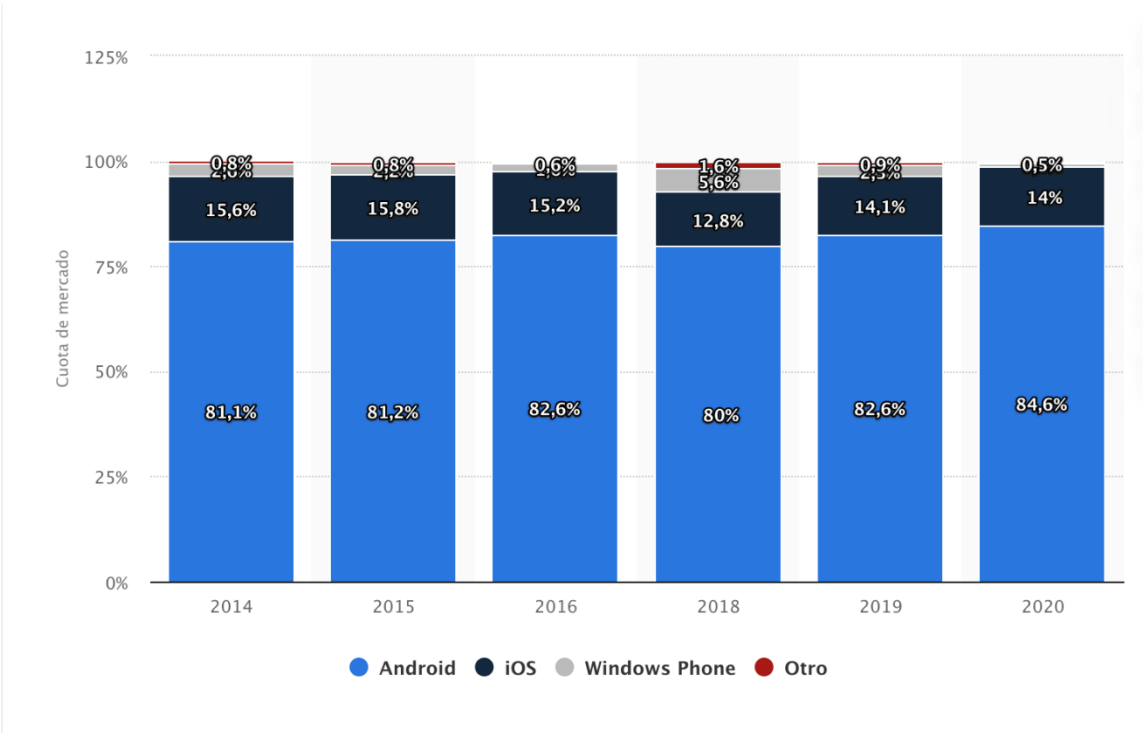


Ilustración 1 Porcentaje de uso de SO en móviles[1]

-Fatiga+PR

Una vez elegido el sistema operativo en Android hay que elegir que versión de Android elegir, por una parte, el elegir una versión muy alta de Android supondrá que se podrá usar en menos dispositivos sin embargo se podrán usar las ultimas actualizaciones y novedades del sistema operativo.

¿Qué versión?

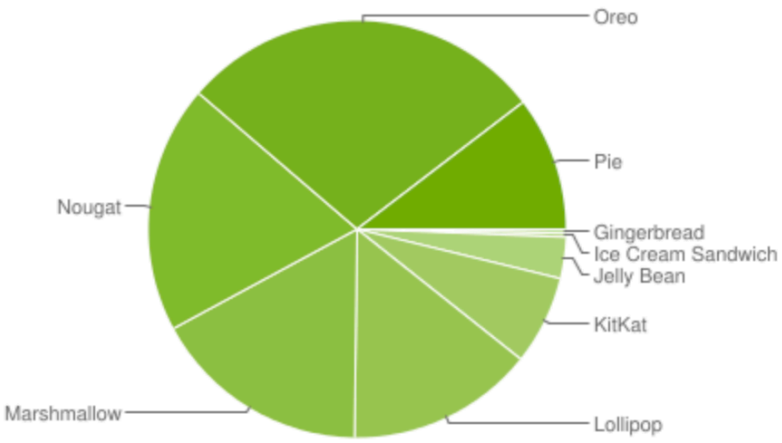
Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.3%
4.1.x	Jelly Bean	16	1.2%
4.2.x		17	1.5%
4.3		18	0.5%
4.4	KitKat	19	6.9%
5.0	Lollipop	21	3.0%
5.1		22	11.5%
6.0	Marshmallow	23	16.9%
7.0	Nougat	24	11.4%
7.1		25	7.8%
8.0	Oreo	26	12.9%
8.1		27	15.4%
9	Pie	28	10.4%

Ilustración 2 Porcentaje de versiones usadas en Android por Android[2]

Se ha elegido realizar aplicación con un SDK mínimo de 21 lo que significa que la aplicación no puede ser usada en un móvil con un sistema operativo de una versión menor a 5.0 o Lollipop.

Usar esta versión te limita a usar esa aplicación en dispositivos Lollipop o superior sin embargo es la versión mínima para usar sistemas de Android como RecyclerView, esenciales para la aplicación, sin embargo según la ultima revisión de Mayo de 2019 el uso de móviles con una versión de Android 5.0 o inferior es de poco superior a 10% por lo que para usar los últimos sistemas de Android merece la pena perder ese 10% de dispositivos mas antiguos que irán desapareciendo con el tiempo. (2)

-Fatiga+PR



Datos recopilados durante un período de 7 días hasta el May 7, 2019.
No se muestran versiones con una distribución inferior al 0.1%.

Ilustración 3 Porcentaje de uso en dispositivos Android actualizados[2]

Por ello la aplicación se podrá usar en el 90% de los móviles Android, lo que significa un 72% de los móviles del parque por lo que es un buen porcentaje de móviles.

3.3 Almacenamiento de datos

Android dispone de varias formas de almacenar datos en la aplicación
El sistema de base de datos que se ha elegido finalmente para almacenar los datos será SQLite, dado que es el tipo de base de datos soportado por el sistema operativo de Android y que tiene los mecanismos incluidos para la modificación de datos de la aplicación. Para el uso de la base de datos SQLite se necesita el paquete **android.database.sqlite.**
Android depende de tres tipos de bases de datos, dependiendo de la volatilidad e importancia de estos es decir dependiendo de si se quiere disponer de ellos siempre que este instalada la aplicación, borrándose cuando se desinstala la aplicación, estos son los llamados shared Preferences, o que sea persistente cuando se desinstale la aplicación como el guardado en SQLite**.**

-Fatiga+PR

SharedPreferences[3]

Se tratan de diccionarios o mapas tipo clave-valor en el que la clave se corresponde a que es que se quiere almacenar, una definición de una palabra única que representa el par clave valor y un valor que como su propio nombre indica es el valor que corresponde a dicha clave. Estos datos quedan guardados hasta que la aplicación es desinstalada o los datos son borrados

Este modo de almacenamiento es el usado en cualquier aplicación de Android bien para almacenar las características correspondientes a la aplicación, como bien puede ser las opciones del sistema operativo de Android, o las opciones particulares de la aplicación, en el caso de esta aplicación se usa para almacenar los datos del usuario tanto nombre como peso y tener una interfaz sencilla para acceder a este par clave valor a la hora de acceder a la base de datos SQLite.

Guardado en el dispositivo

En Android se pueden guardar archivos completos en el dispositivo, estos archivos quedan guardados en el dispositivo.

En ediciones anteriores de Android se permitía el almacenamiento dentro de la memoria interna del dispositivo, sin embargo, el equipo de desarrollo de Android ha ido limitando las exposiciones que puedan tener los usuarios a la memoria principal por razones de seguridad.

El reducir la variedad de maneras de guardar archivos, reduce las probabilidades de encontrar un error.

La reducción este tipo de dualidades ha sido una de las direcciones que ha seguido el equipo de desarrollo de Android desde hace varios años, en cada versión van limitando este tipo de doble posibilidades para mejorar la experiencia de usuario y programadores, para buscar una sola manera controlada y segura. Este tema ha sido discutido varias veces en el podcast Android Developers Podcasts por los propios desarrolladores de Android.

Base de datos SQLite

Android desde sus inicios llevo integrado SQLite en sus dispositivos dado que depende de pocos recursos lo que es ideal para un dispositivo móvil y mas para los antiguos

dispositivos que no tenían mucha capacidad de computo. Tanto es así que SQLite y Room están incluidos en Android Jetpack, librerías base de Android.
Esta forma de guardado de datos es persistente a nuevas instalaciones, es decir, al desinstalar e instalar una aplicación se guardarán los datos de la anterior instalación. Este tipo de almacenamiento se usa para almacenar los ejercicios y el resto de los datos guardados por el usuario en la aplicación. Para ello se usará Room, un asistente de base de datos para Android. “La biblioteca de persistencias Room brinda una capa de abstracción para SQLite que permite acceder a la base de datos sin problemas y, al mismo tiempo, aprovechar toda la potencia de SQLite.”[4].
Room guarda su base de datos SQLite de forma segura y controlada en un directorio reservado de Android, se puede mover al almacenamiento externo, sin embargo, esto no viene sino con otros inconvenientes como que puedes tener un error critico en caso de que la memoria externa se desconecte o deje de funcionar.
La diferencia entre el uso del SQLite sin Room y con Room es que éste está diseñado previamente para seguir el ciclo de vida de una aplicación de Android, que es posiblemente de las partes mas complicadas de controlar de una aplicación Android y sobretodo el ahorro de generar ese trabajo lo convierte en una gran herramienta para Android.

3.4 Persistencia de datos y cambios de estado

Android se compone de actividades y fragmentos principalmente, éstos tienen un ciclo de vida en el dispositivo que es el siguiente:

-Fatiga+PR

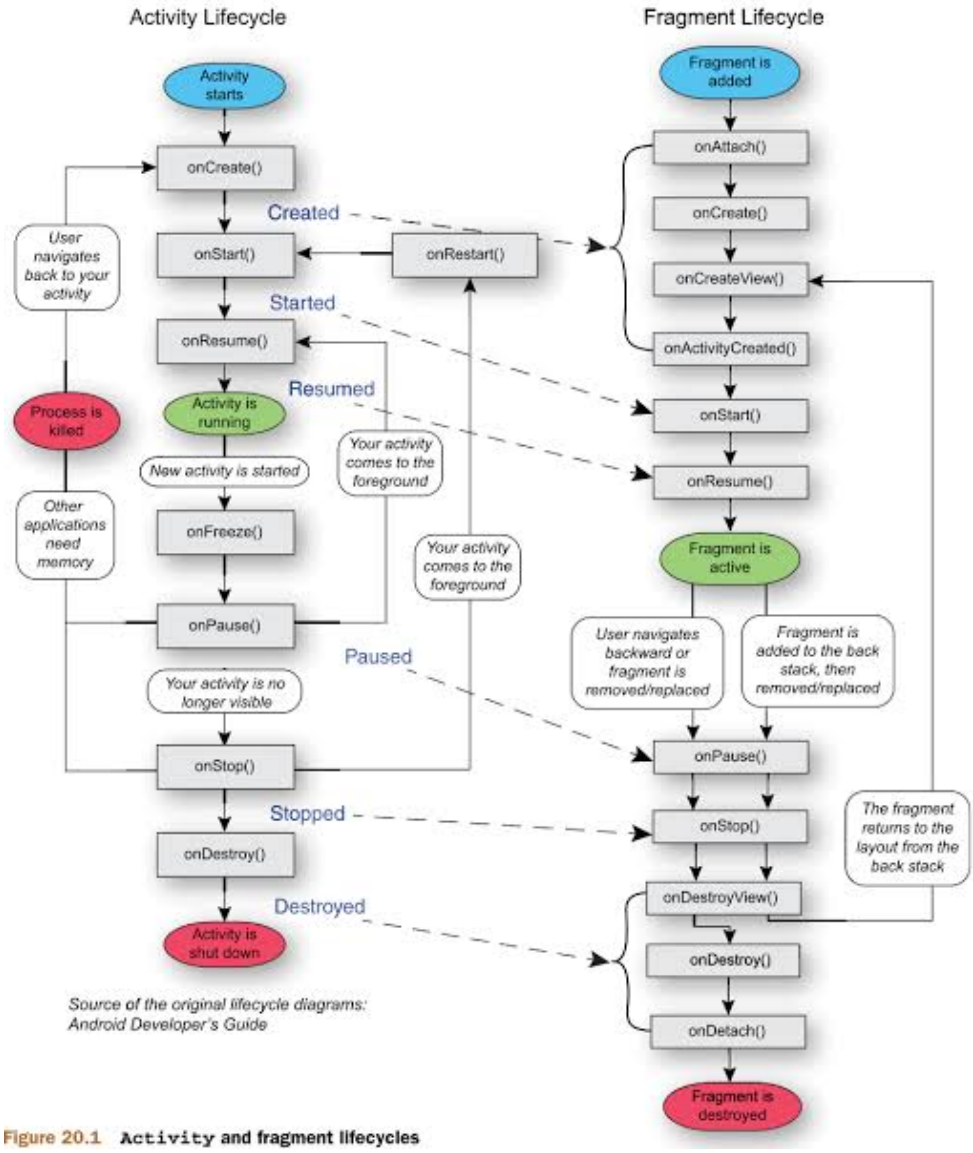


Figure 20.1 `Activity` and fragment lifecycles

Ilustración 4 Ciclo de vida de la actividad y fragmentos[10]

Lo principal en este grafico es que la actividad se destruye volviendo a su estado inicial al realizar acciones como mandar una aplicación a segundo plano, o rotar el teléfono, perdiendo en esta acción toda la información que no se haya guardado se pierde y si no se controla dando fallos de estados en las variables de la aplicación.

Para que una aplicación siga el ciclo de vida de la actividad sin dar lugar a estos errores o estados inconsistentes en Android existen los ViewModels y LiveData que son estructuras que guardan los datos y los cambios de la aplicación “ajenos” al ciclo de vida. En realidad, si que tienen en cuenta el ciclo de vida, de hecho, tiene en cuenta

-Fatiga+PR

dicho ciclo de vida y actúa en consecuencia para que los datos de la aplicación también tengan en cuenta el ciclo de vida. Esta es una de las practicas mas importantes en Android, dado que la rotación y mandar una aplicación a segundo plano es una acción mas que habitual, existen otras formas de mantener la persistencia de datos, pero esta es la manera optima para mucha cantidad de datos.

3.5 RPE

El RPE se trata del esfuerzo máximo percibido de una serie en concreto, una escala que considera el esfuerzo de 1 a 10, siendo 10 no poder hacer más repeticiones en una serie. Es un patrón muy extendido y usado en powerlifting, dado que, a parte de controlar el esfuerzo por serie, valioso para la planificación a largo plazo del entrenamiento, te permite el cálculo de la repetición máxima estimada, es decir, máximo peso que podrías levantar de forma.

3.6 Calculo de volumen en las graficas

El calculo de volumen se usa en powerlifting como un indicador del trabajo realizado en una sesión, cuya formula seria la siguiente:

$$V = \Sigma(p * r)$$

Siendo p el peso que se carga en una serie y la r el numero de repeticiones con ese peso. El sumatorio corresponde a la suma del volumen parcial de cada serie.

3.7 Calculo de la Repetición Máxima en las graficas

El calculo de la repetición máxima es una aproximación a la repetición máxima que se podría haber dado en un día, la mejora de esta estimación implica una mejora implícita en el entrenamiento y posiblemente indicar que se entrena en el camino correcto.
El calculo de esta estimación con la medida de RPE fue “inventada”- introducida desde el atletismo, y adaptada al powerlifting- por Mike Tuchscherer el cual dio a conocer el siguiente grafico con estimaciones entre RPE/Repeticiones para hallar multiplicadores y estimar la Repetición máxima.

-Fatiga+PR

REPS		1	2	3	4	5	6	7	8	9	10	11	12
RPE	10	100.0%	95.5%	92.2%	89.2%	86.3%	83.7%	81.1%	78.6%	76.2%	73.9%	70.7%	68.0%
	9.5	97.8%	93.9%	90.7%	87.8%	85.0%	82.4%	79.9%	77.4%	75.1%	72.3%	69.4%	66.7%
	9	95.5%	92.2%	89.2%	86.3%	83.7%	81.1%	78.6%	76.2%	73.9%	70.7%	68.0%	65.3%
	8.5	93.9%	90.7%	87.8%	85.0%	82.4%	79.9%	77.4%	75.1%	72.3%	69.4%	66.7%	64.0%
	8	92.2%	89.2%	86.3%	83.7%	81.1%	78.6%	76.2%	73.9%	70.7%	68.0%	65.3%	62.6%
	7.5	90.7%	87.8%	85.0%	82.4%	79.9%	77.4%	75.1%	72.3%	69.4%	66.7%	64.0%	61.3%
	7	89.2%	86.3%	83.7%	81.1%	78.6%	76.2%	73.9%	70.7%	68.0%	65.3%	62.6%	59.9%
	6.5	87.8%	85.0%	82.4%	79.9%	77.4%	75.1%	72.3%	69.4%	66.7%	64.0%	61.3%	58.6%

Ilustración 5 Correlación entre RPE y porcentaje de RM[9]

Siguiendo la formula:

$$RM = p/(\frac{fc}{100})$$

Siendo p el peso de la serie y fc el factor de conversión de la tabla anterior.

-Fatiga+PR

Técnicas y Herramientas

4.1 Herramienta de Control de Versiones

Se utiliza GitHub como herramienta de control de versiones dado que es una herramienta previamente usada y con cierta familiaridad.
GitHub aloja tu repositorio de código y te brinda **herramientas** muy útiles para el **trabajo en equipo**, dentro de un proyecto.[5]

4.2 Herramienta de Gestión de proyectos

ZenHub es una herramienta de gestión directamente integrada en GitHub, que usa la información de cada Issue para organizar el proyecto en tiempo real para una mejor interacción con las fases del proyecto.
Se ha elegido ZenHub por la familiaridad y su integración con GitHub

4.3 Gestor de Referencias bibliográficas

Zotero es una extensión libre para el navegador Firefox, que permite a los usuarios recolectar, administrar y citar investigaciones de todo tipo.[6]

4.4 Lenguaje de programación

Se elegirá kotlin para el desarrollo de la aplicación dado que desde la convención de

-Fatiga+PR

google I/O de 2017 kotlin es uno de los lenguajes oficiales de Android, y se esta convirtiendo en el lenguaje mas usado por propios desarrolladores de mejoras de Android, por ello se puede presuponer que seguirá creciendo en el futuro.
Kotlin es un lenguaje de programación basado en java, es interoperable con java, lo que te permite llamar a librerías creadas en java sin problema ninguno.
Tiene ciertas diferencias con java y según van avanzando las versiones kotlin se va o intenta ir alejando de java, en este punto es un lenguaje tan importante en Android que Android dejo de soportar versiones de java superiores a 8 mientras que siempre esta al día de versiones de kotlin.
Kotlin se empezó a popularizar y distinguir de java por el hecho de que su manera de controlar las excepciones por nulo es decir las NullPointerException.
Sin embargo, desde estas primeras diferencias kotlin se ha ido diferenciando cada vez mas de java.
Las principales características y elementos que hacen diferencian a kotlin:

- **Corrutinas**: Principal elemento por el cual kotlin ha sido el elegido por Android para ser su lenguaje referente. Android esta construido por su esencia para dispositivos con recursos limitados y reducidos, por ello el uso de Corrutinas. Corrutinas se corresponde a un proceso asíncrono que no usa un hilo propio para realizar esta tarea asíncrona, por lo que para dispositivos sin varios hilos como los móviles mas antiguos es ideal y a los nuevos también les mejora su capacidad de computo por el hecho de dejar un hilo libre para el resto de los procesos.
- **Síntesis de código**: Kotlin te permite reducir las líneas de código de java, como ejemplo puedes reducir una clase de java a una línea con todos los getters y los setters de la clase ya construidos.

4.5 Librerías

Las librerías usadas en esta aplicación para poder usar herramientas especificas en Android se nombrarán en este apartado, se dividirán en librerías de Android y librerías externas.

4.5.1 Librerías Android

-Fatiga+PR

Aquí están incluidas todas las librerías que estén desarrolladas o supeditadas por el equipo de desarrollo de Android.

4.5.1.1 Androidx RecyclerView

Librería usada para poder trabajar con listas de datos con la técnica RecyclerView, técnica mas usada para la visualización de datos en una ordenación concreta dado que permite un mejor uso de la memoria dado que los elementos de la lista se crean y destruye dinámicamente cuando aparece o desaparece de la visión del usuario.

4.5.1.2 Androidx Room

Librería usada con el fin del uso de los mecanismos de Room para el facilitar el uso de la base de datos SQLite como se ha redactado anteriormente.

4.5.1.3 Androidx Espresso

Librería usada con el fin del testear de la aplicación.

4.5.1.4 Androidx Firebase

Librería usada con el fin de poder conectar la aplicación con recursos en línea con los recursos de FireBase, en esta aplicación se usará para iniciar la sesión con un usuario y guardarlo en línea.

4.5.2 Librerías externas

Estas son las librerías que no pertenecen al equipo de desarrolladores de Android

4.5.2.1 MPAndroidChart

Librería usada con el fin de realizar graficas dentro de Android.

-Fatiga+PR

4.5.2.2 Robolectric

Librería usada para mejorar la manera de leer las pruebas de Espresso.

-Fatiga+PR

Aspectos relevantes del desarrollo

5.1 Metodología

Para realizar este proyecto se hizo el uso de la tecnología scrum, a pesar de que se adaptara a las características del proyecto, dado que es individual, por ejemplo.
En la metodología Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Se basa en:

- El desarrollo incremental de los requisitos del proyecto en bloques temporales cortos y fijos
- El desarrollo empírico del proyecto, se comprueba el funcionamiento realizado en ese sprint
- El timeboxing -disponer de un tiempo máximo para realizar una tarea- de las actividades del proyecto, para ayudar a la toma de decisiones y conseguir resultados.

5.2 Formación

Para realizar este proyecto se ha requerido de muchos recursos online dado que en el grado tanto el lenguaje de kotlin como las aplicaciones para el sistema operativo Android no se han estudiado, a pesar de que Kotlin no ha sido necesario mas que búsquedas puntuales a StackOverflow.

- Developing Android Apps
- FireBase in a Weekend
- Advance Android with Kotlin

5.3 Arquitectura MVVM

-Fatiga+PR

Es acrónimo de Model-View ViewModel[7]:

- **View**: Se trata de la interfaz del dispositivo. Observa el ViewModel, lo que quiere decir que se le notifica cuando el ViewModel registra un cambio en el Model.
- **Model**: Se trata de los datos intrínsecos de la aplicación. Dispone de elementos tipo observables que notifican al ViewModel cuando se produce un cambio en la aplicación.
- **ViewModel**: Intermediario entre View y Model. Transforma los datos del Model para proporcionárselos a la View. Para la comunicación View ViewModel se abren canales da datos.

Diferencias entre MVVM y MVC

A simple vista MVVM y MVC pueden parecer muy similares, pero existen algunas diferencias:

- El Controlador tiene referencias a la View sin embargo el ViewModel no.
- El controlador actualiza la UI de manera clásica mientras que el ViewModel con canales de datos.
- Un mismo ViewModel puede actualizar varias View mientras que en MVC el cada Vista tiene su propio Controlador.
- Binder: Se trata de un elemento de unión entre ViewModel y View que permite un mejor aislamiento de la UI.[8]

-Fatiga+PR

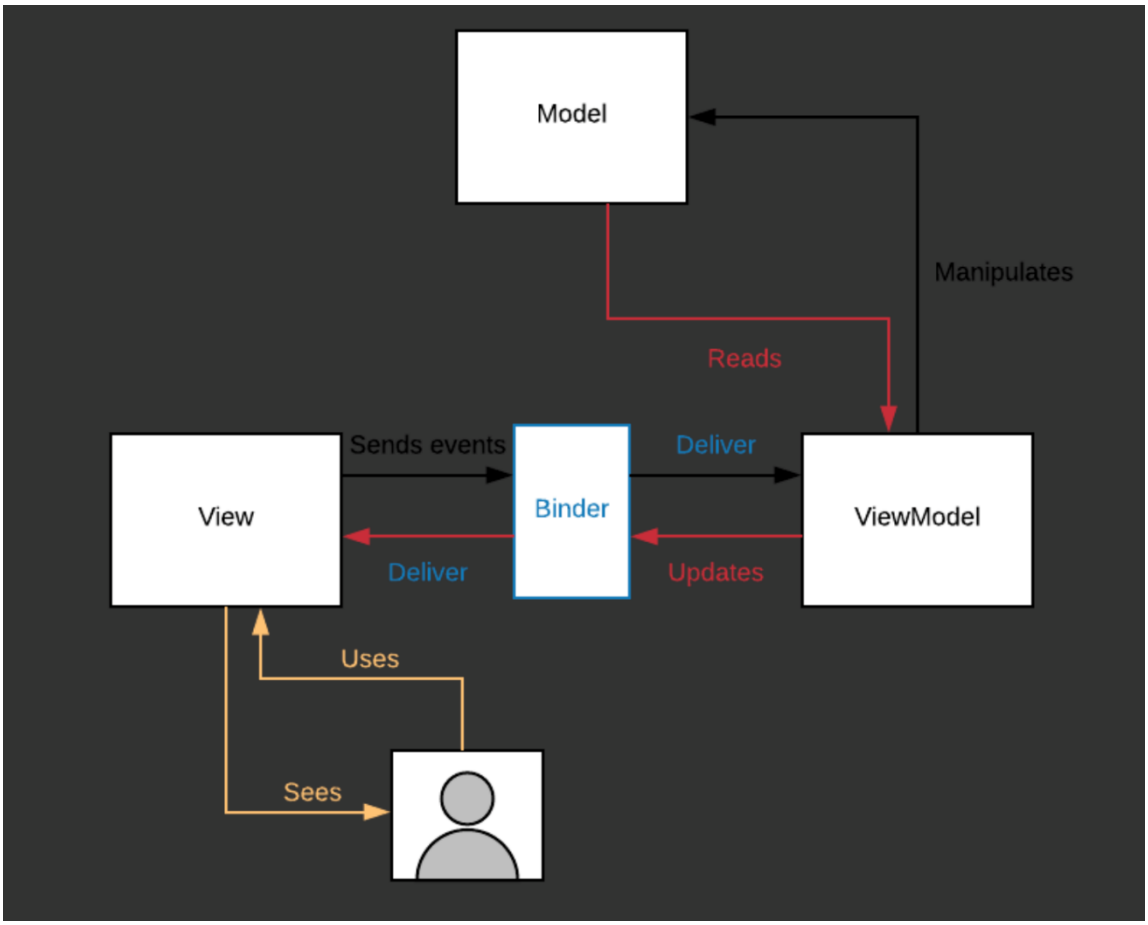


Ilustración 6 Estructura de MVVM[8]

Básicamente MVVM es una mejora del MVC dado que permite una mayor capacidad de testeo y separa fielmente la View del elemento intermediario (ViewModel o Controlador)

-Fatiga+PR

Trabajos relacionados

Al inicio del proyecto se realizo un estudio de algunas de las aplicaciones del sector, estos fueron los resultados

Funcionalidad\Aplicacion	GymRun	BestFit	GymDaily	FitNotes	MyWorkoutPlan	GymWP
Ejercicios por partes del cuerpo	x	Lista	x	x	x	x
Rutina Usada	Predeterminada	Propia	Propia	Propia	Propia	Ambas
Link video explicativos	Youtube	Integrado				
Introduccion datos ejercicio	x	x	x	x	x	x
Datos corporales	x	x			x	x
Graficos	x	x		x		x
Historial	x		x	x	x	x
Usuarios	Varios					Uno(Opcion login)
Smartwatch	x					
Calendario	x		x	x	x	
Opcion de limitacion de uso de datos		x				
Calculo RM			x	x		
Seleccion de unidad de medida				x	x	
Opcion de no bloquear pantalla				x	x	
Copia de seguridad en la nube				x		
Creacion de ejercicio propio					x	
Niveles de fatiga estaticos						x

Una vez finalizado el proyecto se puede comparar con el resto de las aplicaciones de la AppStore.

Funcionalidad\Aplicacion	GymRun	BestFit	GymDaily	FitNotes	MyWorkoutPlan	GymWP	-Fatiga+PR
Ejercicios por partes del cuerpo	x	Lista	x	x	x	x	x
Rutina Usada	Predeterminada	Propia	Propia	Propia	Propia	Ambas	Predeterminada
Link video explicativos	Youtube	Integrado					
Introduccion datos ejercicio	x	x	x	x	x	x	x
Datos corporales	x	x			x	x	x
Graficos	x	x		x		x	x
Historial	x		x	x	x	x	x
Usuarios	Varios					Uno(Opcion login)	Uno Consecutivo
Smartwatch	x						
Calendario	x		x	x	x		x
Opcion de limitacion de uso de datos		x					
Calculo RM			x	x			x
Seleccion de unidad de medida				x	x		
Opcion de no bloquear pantalla				x	x		
Copia de seguridad en la nube				x			
Creacion de ejercicio propio					x		
Niveles de fatiga estaticos						x	

Se puede ver como cumple con la mayoría de las especificaciones de las aplicaciones.

-Fatiga+PR

Hay una cosa que puntualizar y es que el numero de ejercicios que existen en el resto de las aplicaciones es mayor, esto se explica porque este proyecto esta especializado en powerlifting, que solo utiliza tres ejercicios básicos.

Conclusiones y Líneas de trabajo futuras

Introducción

En este apartado se verán que conclusiones se han sacado de la realización del proyecto y que elementos del mismo se pueden mejorar en el futuro

Conclusiones

En cuanto a los objetivos del proyecto se puede considerar que se han cumplido los objetivos del proyecto iniciales. Se ha conseguido una aplicación plenamente funcional para el entrenamiento especifico de powerlifting. Además con el conocimiento adquirido en Android se ha conseguido ir mas de lo esperado en la dirección de mejora de utilización de memoria y uso de estructuras particulares de aplicaciones en Android.

En cuanto a objetivos personales también se consideran cumplidos, se inició el proyecto con conocimientos mínimos de Android y Kotlin y se ha conseguido realizar una aplicación funcional, además se ha realizado una investigación de muchos aspectos de Android, pudiendo realizar una aplicación siguiendo los criterios que los desarrolladores de Android consideran como las guías correctas de programación y de diseño.

Líneas de trabajo futuras

En este apartado se verán los apartados o funcionalidades nuevas que se pueden introducir en la aplicación

Opciones de la aplicación

Aquí se pondrán las mejoras que corresponden con elementos que no añaden

funcionalidad, sino que seria parte de conocimiento de Android.

Personalización de la interfaz

En esta opción se intentaría tener colores personalizados para la aplicación así como una opción de modo noche

Funcionalidades de la aplicación

Aquí se pondrán las mejoras que añadan funcionalidad a la aplicación

Añadir una opción entrenador

Seria un buen movimiento el hecho de tener dos tipos de usuarios, uno que sea atleta y otro el entrenador.

Este nuevo usuario podría tener acceso a los entrenamientos de varios atletas a los cuales va a programar e introducir sus ejercicios.

Añadir la opción de planificación

Añadir además de las series que se han hecho las que se pretenden hacer el día del entreno, es decir planificar previamente el entrenamiento

Introducir toda la base de datos en línea

Esta opción podría subir toda la base de datos de todos los usuarios a un servidor de FireBase, probablemente esta opción tenga un costo monetario de alquiler de almacenamiento

Introducir la opción de videos

Introducir videos de los levantamientos que te redirijan a YouTube, una mejora a esta opción seria que ni siquiera se salga de la aplicación y que se abra en una pestaña sobre la aplicación.

Personalización de la tabla para calculo de la RM

Una opción que te permita mejorar la tabla de RM, la tabla usada en esta aplicación es una tabla genérica, que si bien cumple su función tener la capacidad de personalización es un plus.

-Fatiga+PR

Función SmartWatch

En esta opción se podría hacer un seguimiento del entrenamiento desde el smartwatch que evita sacar el móvil en mitad del entrenamiento

-Fatiga+PR

Bibliografía

[1] Statista. Cuota de mercado de pedidos de Smartphone [30 de abril de 2020]

[2] Android Developers. Versiones de Android.[1 de mayo de 2020]

[3] Android Developer. Shared Preferences[1 de mayo de 2020]

[4] Android Developer. Room [1 de mayo de 2020]

[5] Conociendo GitHub. Introducción a. GitHub[3 de mayo de 2020]

[6] Zotero. Zotero[2 de octubre de 2019]

[7] JournalDev. Patrón Mvvm. [15 de mayo de 2020]

[8] RayWndelich. Patrón Mvvm [15 de mayo de 2020]

[9] RTS. Cuadro RPE. [15 abril 2020]

[10] Paintrest. Ciclo de Vida de Android Completo [30 de junio de 2020]

-Fatiga+PR