



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA (SG)
Grado en Ingeniería Informática de Servicios y Aplicaciones

RefWatch

Aplicación Android y Android Wear de asistencia de árbitros



Alumno: Daniel Callejo Matey

Tutor: Luis Ignacio Sebastián Martín

Agradecimientos

Muchas gracias a todos aquellos que han compartido estos años de carrera conmigo, ha sido una etapa muy importante y con vosotros ha sido aún mejor.

Gracias a toda mi familia por apoyarme todos estos años y me ha ayudado tanto.

Gracias a mi pareja por estar siempre ahí para ayudar cuando se la necesitaba.

Gracias a todos los profesores por los conocimientos impartidos a lo largo de la carrera que me han permitido llegar a donde estoy hoy en día y en especial a mi tutor Luis Ignacio Sebastián Martín por apoyarme a lo largo de este TFG.

Resumen

El objetivo de este proyecto es el desarrollo de una aplicación wear y móvil para árbitros de fútbol, con el propósito de facilitar la creación de actas de los partidos de fútbol.

Se tratará de facilitar la tarea del árbitro a lo largo del partido, gestionando el tiempo y las distintas incidencias que transcurran durante el periodo del partido.

Abstract

The objective of this project is to develop an application for Wear and Android devices for soccer referees in order to help the referee with the creation of the minute after the match.

It will be about to easy the task of the referee along the game, managing the time and the different incidences carried out during the match period.

Contenido

Capítulo 1 Introducción	11
1.1. Motivación	11
1.2. Objetivos y alcance	12
1.3. Organización de la memoria	14
1.4. Contenido del DVD	15
Capítulo 2 Estado del arte.....	17
2.1. IOS	17
2.2. Android	20
2.3. Comparativa.....	23
Capítulo 3 Planificación y presupuesto	25
3.1. Metodología	25
3.2. Estimación temporal	25
3.3. Estimaciones	28
3.3.1. Puntos de función	28
3.3.2. COCOMO 2	33
3.4. Presupuesto	35
3.4.1. Presupuesto inicial.....	35
3.4.2. Presupuesto real.....	37
3.5. Conclusiones	38
Capítulo 4 Análisis	39
4.1. Requisitos de usuario	39
4.1.1. Descripción de los actores.....	39
4.1.2. Listado de requisitos de usuario	40
4.1.3. Diagramas de casos de uso.....	42
4.1.4. Especificación de casos de uso.....	44
4.2. Requisitos de información	51
4.2.1. Modelo de datos conceptual.....	52
4.2.2. Diccionario de datos.....	53
Capítulo 5 Diseño	55
5.1. Arquitectura lógica	55
5.2. Arquitectura física.....	60
5.3. Diagrama de secuencia	61
5.4. Modelo lógico de datos.....	63

5.5. Interfaz de usuario	63
Capítulo 6 Implementación	73
6.1. Herramientas de análisis	73
6.2. Herramientas de desarrollo	77
Capítulo 7 Pruebas.....	87
7.1. Pruebas de caja blanca	87
7.2. Pruebas de caja negra.....	92
Capítulo 8 Manuales.....	97
8.1. Manual de instalación	97
8.2. Manual de uso.....	98
Capítulo 9 Conclusiones.....	105
9.1. Posibles ampliaciones	105
Referencias bibliográficas	107
Anexos.....	109

Índice de tablas

Tabla 1 Comparativa de aplicaciones	23
Tabla 2 Complejidad según parámetros	28
Tabla 3 LDC por PF	29
Tabla 4 Complejidad de componentes de aplicación móvil	29
Tabla 5 Tabla de puntos de función sin ajustar de aplicación móvil.....	30
Tabla 6 Coeficientes para el factor de ajuste de aplicación móvil	30
Tabla 7 Complejidad de componentes de aplicación Wear.....	32
Tabla 8 Coeficientes para el factor de ajuste de aplicación Wear	32
Tabla 9 Coeficientes para el factor de ajuste de aplicación Wear	33
Tabla 10 Modelos de COCOMO.....	33
Tabla 11 Factores de ajuste de COCOMO	34
Tabla 12 Elección de factores de ajuste de COCOMO	34
Tabla 13 Presupuesto hardware	35
Tabla 14 Presupuesto software	36
Tabla 15 Presupuesto de personal	36
Tabla 16 Presupuesto total.....	36
Tabla 17 Presupuesto hardware real	37
Tabla 18 Presupuesto software real	37
Tabla 19 Presupuesto de personal real	38
Tabla 20 Presupuesto total real.....	38
Tabla 21 Actores aplicación móvil.....	39
Tabla 22 Actores aplicación Wear.....	39
Tabla 23 Requisitos de usuario aplicación móvil.....	40
Tabla 24 Requisitos de usuario aplicación Wear.....	41
Tabla 25 CU-02 Configurar equipos	45
Tabla 26 CU-03 Guardar partido pendiente	45
Tabla 27 CU-04 Mostrar partidos pendientes	46
Tabla 28 CU-07 Empezar partido pendiente	47
Tabla 29 CU-13 Terminar partido	48
Tabla 30 CU-13 Iniciar/pausar cronómetro.....	49
Tabla 31 CU-15 Cambiar periodo	50
Tabla 32 CU-16 Añadir gol	51
Tabla 33 Entidad Partido	53
Tabla 34 Entidad Equipo	53
Tabla 35 Entidad Jugador	53
Tabla 36 Entidad Incidencia	54
Tabla 37 Portada Aplicación móvil	65
Tabla 38 Pantalla configurar partido	66
Tabla 39 Pantalla configurar equipos	67
Tabla 40 Pantalla configurar titulares.....	68
Tabla 41 Pantalla de espera de partido	69
Tabla 42 Pantalla principal de Android Wear	70
Tabla 43 Pantalla añadir gol	71

Tabla 44 Pantalla añadir tarjeta	72
Tabla 45 Prueba de caja blanca 1	89
Tabla 46 Prueba de caja blanca 2	89
Tabla 47 Prueba de caja blanca 3	89
Tabla 48 Diagrama de flujo de caja blanca 2	91
Tabla 49 Prueba de caja blanca 1	92
Tabla 50 Prueba de caja blanca 2	92
Tabla 51 Prueba de caja blanca 3	92
Tabla 52 Prueba de caja blanca 4	92
Tabla 53 Prueba de caja blanca 5	92
Tabla 54 Prueba de caja blanca 6	92
Tabla 55 PN-01: Configurar Partido	93
Tabla 56 PN-02: Configurar Equipos	93
Tabla 57 PN-03: Ver partidos pendientes.....	94
Tabla 58 PN- 04: Recibir partido	94
Tabla 59 PN-05: Iniciar/pausar cronómetro	95
Tabla 60 PN-06: Añadir gol	95
Tabla 61 CU-01 Configurar partido	114
Tabla 62 CU-02 Configurar equipos	115
Tabla 63 CU-03 Guardar partido pendiente	116
Tabla 64 CU-04 Mostrar partidos pendientes	117
Tabla 65 CU-05 Mostrar partido pendiente.....	118
Tabla 66 CU-06 Borrar partido pendiente	119
Tabla 67 CU-07 Empezar partido pendiente	120
Tabla 68 CU-08 Recibir partido terminado	121
Tabla 69 CU-09 Mostrar partidos terminados	122
Tabla 70 CU-10 Mostrar partido terminado	123
Tabla 71 CU-11 Borrar partido terminado	124
Tabla 72 CU-12 Recibir partido	125
Tabla 73 CU-13 Terminar partido	126
Tabla 74 CU-13 Iniciar/pausar cronómetro.....	127
Tabla 75 CU-15 Cambiar periodo	128
Tabla 76 CU-16 Añadir gol	129
Tabla 77 CU-17 Añadir substitución.....	130
Tabla 78 CU-18 Añadir tarjeta	131
Tabla 79 CU-19 Actualizar cronómetros	132
Tabla 80 CU-20 Finalizar cronómetros	133

Índice de ilustraciones

Ilustración 1.1 Android Vs IOS (ComTech, 2018)	12
Ilustración 1.2 Árbol de características móvil	13
Ilustración 1.3 Árbol de características Wear	14
Ilustración 2.1 Icono Crono Fútbol (2011)	17
Ilustración 2.2 Capturas Crono Fútbol	17
Ilustración 2.3 Icono Goref	18
Ilustración 2.4 Capturas Goref móvil	18
Ilustración 2.5 Capturas Goref Wear	19
Ilustración 2.6 Capturas Cronómetro De Fútbol APK	20
Ilustración 2.7 Capturas Soccer Referee Pro	21
Ilustración 2.8 Capturas Refsix	22
Ilustración 3.1 Planificación temporal	26
Ilustración 3.2 Diagrama de Gantt	27
Ilustración 4.1 Diagrama de casos de uso de aplicación móvil	42
Ilustración 4.2 Diagrama de casos de uso de aplicación Wear	43
Ilustración 4.3 Modelo Entidad-Relación	52
Ilustración 5.1 Arquitectura lógica	55
Ilustración 5.2 Dependencias Arquitectura Lógica (Cejas, 2014)	56
Ilustración 5.3 Capa de presentación	57
Ilustración 5.4 Modelos y mappers de la capa de presentación	57
Ilustración 5.5 Capa de dominio	58
Ilustración 5.6 Modelos y mappers	58
Ilustración 5.7 Capa de data	59
Ilustración 5.8 Room Database	59
Ilustración 5.9 Provide connect repository	59
Ilustración 5.10 Arquitectura física	60
Ilustración 5.11 Diagrama de secuencia Guardar partido pendiente	61
Ilustración 5.12 Diagrama de secuencia Mostrar partidos	62
Ilustración 5.13 Diagrama de secuencia Iniciar / pausar partido	62
Ilustración 5.14 Modelo Relacional	63
Ilustración 5.15 Problemas de layout en Android Wear (Google, 2018)	64
Ilustración 5.16 Arreglar problema de layout de Android Wear (Google, 2018)	64
Ilustración 5.17 Portada aplicación móvil	65
Ilustración 5.18 Pantalla configurar partido	66
Ilustración 5.19 Pantalla configurar equipos	67
Ilustración 5.20 Pantalla configurar titulares	68
Ilustración 5.21 Pantalla de espera de partido	69
Ilustración 5.22 Pantalla principal de Android Wear	70
Ilustración 5.23 Pantalla añadir gol	71
Ilustración 5.24 Pantalla añadir tarjeta	72
Ilustración 6.1 Word	73
Ilustración 6.2 Draw.io	74
Ilustración 6.3 Android Studio	74

Ilustración 6.4 Máquinas Virtuales de Android Studio	75
Ilustración 6.5 SimpleUMLCE.....	76
Ilustración 6.6 Uso de SimpleUMLCE	76
Ilustración 6.7 Tomsplanner.....	77
Ilustración 6.8 Room database (Ajay, 2017)	78
Ilustración 6.9 Android Debug Database	79
Ilustración 6.10 Wear OS (Google, s.f.).....	80
Ilustración 6.11 Start Espresso test.....	85
Ilustración 6.12 Grabar test con Espresso	85
Ilustración 7.1 Diagrama de flujo de caja blanca 1	88
Ilustración 8.1 Activar opciones de desarrollo en Android Wear (Subirats, 2014).....	97
Ilustración 8.2 Activar depuración WIFI Android Wear.....	98
Ilustración 8.3 Aplicación Android Wear.....	99
Ilustración 8.4 Pantalla principal de la aplicación móvil.....	99
Ilustración 8.5 Pantalla de configuración de partidos.....	100
Ilustración 8.6 Pantalla de configuración de equipos	100
Ilustración 8.7 Pantalla de selección de jugadores	101
Ilustración 8.8 Partidos pendientes	101
Ilustración 8.9 Información sobre un partido	102
Ilustración 8.10 Pantalla principal de la aplicación Wear	102
Ilustración 8.11 Menú de aplicación Wear	103
Ilustración 8.12 Añadir gol.....	103
Ilustración 8.13 Añadir tarjeta.....	104
Ilustración 8.14 Partido finalizado	104

Capítulo 1 Introducción

En este apartado se explicarán la motivación para realizar este proyecto para Trabajo de Fin de Grado y los objetivos y alcance del mismo.

1.1. Motivación

Las aplicaciones móviles siempre me han parecido interesantes, y la posibilidad de poder desarrollar una aplicación móvil y wear llamó mi atención hacia este proyecto.

El deporte siempre me ha gustado y me ha parecido que se podría utilizar gran variedad de tecnologías dentro de ellos, actualmente en el deporte se están empezando a utilizar más tecnologías como es el caso del ojo de halcón en el fútbol para determinar cuando el balón ha sobrepasado la línea de gol, o actualmente el video arbitraje (VAR) utilizado en el mundial.

Y aun cuando la tecnología ha mejorado, los árbitros siguen teniendo que apuntar las incidencias del partido a mano, lo cual tiene sus inconvenientes:

- Cuando llueve se complica la tarea de anotar las incidencias.
- Gran tiempo para redactar el acta tras el partido.
- Poco control del tiempo perdido por pausas durante el partido.

En la actualidad, con las facilidades que nos ofrecen los Smartphone y con la reciente creación de Android Wear en 2014, se podría ayudar al árbitro en su tarea de realizar el seguimiento de las incidencias.

Con el objetivo de desarrollar la aplicación, nos planteamos para que sistema operativo sería más apropiado realizarla.

Se nos plantean dos opciones:

- Sistema operativo IOS y Apple Watch.
- Sistema operativo Android y Wear OS.

Actualmente en España las cifras de usuarios de Android respecto a usuarios de IOS revelan que Android es más utilizado por los españoles.

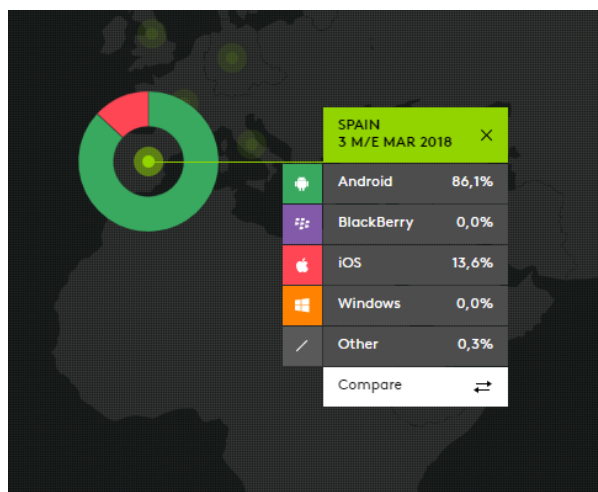


Ilustración 1.1 Android Vs IOS (ComTech, 2018)

Como podemos observar en la ilustración 1.1 el 86.1% de los usuarios de teléfonos móviles en España dispone de un dispositivo Android, mientras IOS solo es usado por un 13.6% de la población.

Debido al mayor número de usuarios de Android decidimos desarrollar nuestra app para Android, con el objetivo de facilitar su uso a un mayor número de personas.

1.2. Objetivos y alcance

El objetivo principal de este TFG es el desarrollo de una aplicación Android y wear para facilitar la labor de los árbitros durante el partido, para ello se toma como punto de partida la aplicación de Goref de IOS mencionada más adelante en el apartado de estado del arte.

Para facilitar estas tareas buscaremos:

- La posibilidad de configurar los partidos previamente, ofreciendo de esta forma la posibilidad de elegir los dorsales de los jugadores que disputaran el partido y los nombres de los equipos.
- Manejo de tiempos del partido con distintos cronómetros para facilitar la gestión del tiempo de juego y las pausas transcurridas durante el encuentro.
- Creación de incidencias, ya sean en forma de goles, sustituciones o tarjetas que permiten al árbitro reportar lo transcurrido a lo largo del encuentro asignando las incidencias a un jugador y tiempo concreto.
- Visualización de los datos del partido una vez terminado desde la aplicación móvil.

Distinguiendo la aplicación móvil y wear podemos apreciar distintas características según el sistema.

Por un lado la aplicación móvil constará de las siguientes características principales como podemos observar en la ilustración 1.2:

- Gestión de partidos: permitirá la gestión de partidos en la aplicación ya sea la configuración de partidos, o la visualización de partidos ya terminados con sus diferentes incidencias.
- Gestión de conexión: permitirá el envío de partidos del teléfono al reloj para iniciar los partidos o del reloj al teléfono para guardar la información del partido una vez este ha sido finalizado.

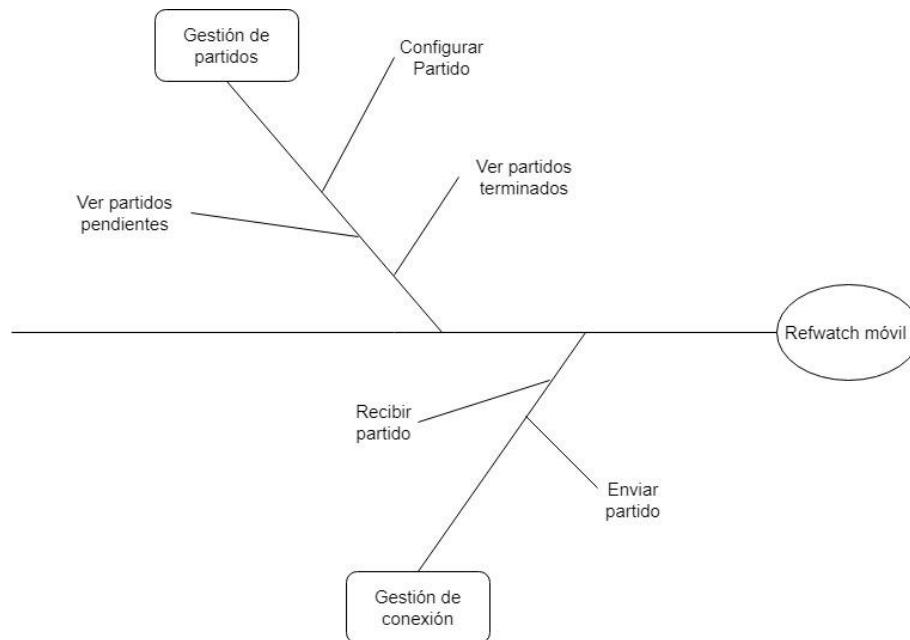


Ilustración 1.2 Árbol de características móvil

La aplicación wear, por otro lado, constará de las siguientes características como podemos observar en la ilustración 1.3:

- Gestión del tiempo: permite la gestión del tiempo a lo largo del partido con distintos cronómetros y distintos periodos de tiempo.
- Gestión de incidencias: permite la gestión de incidencias ya sean goles sustituciones o tarjetas.
- Gestión de conexión: permitirá la recepción de partidos del teléfono al reloj para empezar el partido y el envío de partidos del reloj al teléfono para finalizar el encuentro y registrar las incidencias del partido.

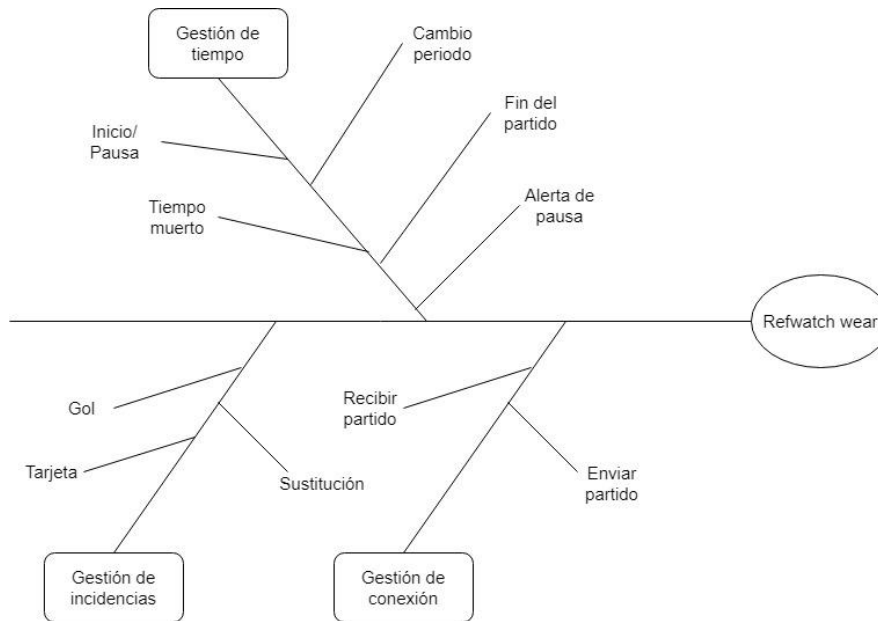


Ilustración 1.3 Árbol de características Wear

1.3. Organización de la memoria

En este apartado se describen los diferentes capítulos de este documento. La memoria contará con nueve capítulos con sus respectivas secciones y un apartado de. A continuación se realizará una breve descripción del contenido de cada uno de los apartados de la memoria.

- **Capítulo 1 Introducción:** En este apartado se explicarán la motivación para realizar este proyecto para Trabajo de Fin de Grado y los objetivos y alcance del mismo.
- **Capítulo 2 Estado del arte:** Existen varias aplicaciones, ya sean para IOS o para Android que poseen funcionalidades similares a la de este Trabajo de Fin de Grado buscando facilitar la labor de los árbitros.
- **Capítulo 3 Planificación y presupuesto:** En este capítulo se hablará sobre la metodología, la planificación y la estimación de presupuesto del desarrollo de este proyecto.
- **Capítulo 4 Análisis:** En este capítulo se muestra el análisis previo a la implementación de ambos sistemas, Android y Android Wear, que se van a desarrollar. Al ser dos sistemas distintos, las especificaciones de estos estarán separadas cuando sea requerido
- **Capítulo 5 Diseño:** En este capítulo se expone el diseño que se tendrá en cuenta a la hora de desarrollar el sistema. Se tratarán las arquitecturas lógicas y físicas, el diagrama de clases, diagramas de secuencias representativos del sistema, el modelo lógico de datos y los diseños de algunas de las pantallas de interfaz de usuario del sistema.

- **Capítulo 6 Implementación:** A lo largo de este capítulo se explicarán las diferentes herramientas utilizadas para el desarrollo y análisis de los sistemas Android wear y Android.
- **Capítulo 7 Pruebas:** En este capítulo se describirán las pruebas realizadas sobre los dos sistemas. Se han realizado dos tipos de pruebas, pruebas de caja blanca y pruebas de caja negra.
- **Capítulo 8 Manuales:** En este capítulo se explicará cómo realizar la instalación de los sistemas, tanto el Android Wear como el Android, y de cómo utilizar la aplicación.
- **Capítulo 9 Conclusiones:** En este capítulo se reflexionará sobre la elección del TFG, su desarrollo, aprendizaje durante el proceso y posibles mejoras que se podrían realizar sobre dicho proyecto en futuras versiones.

1.4. Contenido del DVD

El DVD adjuntado contiene los siguientes elementos:

- Wear.apk: instalador de la aplicación Android Wear.
- Android.apk: instalador de la aplicación Android.
- refwatch: carpeta con el proyecto de la aplicación de Android y la aplicación Android Wear.
- Memoria-DCM.pdf: pdf con este documento sobre la memoria del proyecto.

Capítulo 2 Estado del arte

Existen varias aplicaciones, ya sean para IOS o para Android que poseen funcionalidades similares a la de este TFG buscando facilitar la labor de los árbitros.

A continuación se mostraran las aplicaciones destacadas en ambos sistemas operativos

2.1. IOS

Crono Fútbol

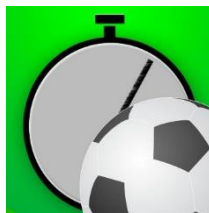


Ilustración 2.1 Icono Crono Fútbol (2011)

Aplicación para fanáticos del fútbol que permite llevar el tiempo del partido para saber en todo momento el minuto de juego.

Permite configurar de manera personalizada la duración de cada periodo y de las prórrogas, así con la gestión de tarjetas en su versión de pago por 2.29€.

Para la visualización del tiempo se puede elegir entre visualización progresiva o regresiva.

En las siguientes capturas podemos apreciar algunas de las pantallas de la aplicación. En la primera captura podemos observar la configuración del tiempo del partido. En la segunda pantalla se muestra el inicio del partido. En la tercera se puede ver la aplicación durante el transcurso del partido, durante la cual se puede pausar detener o cancelar el partido, además de ver la información referente al encuentro.

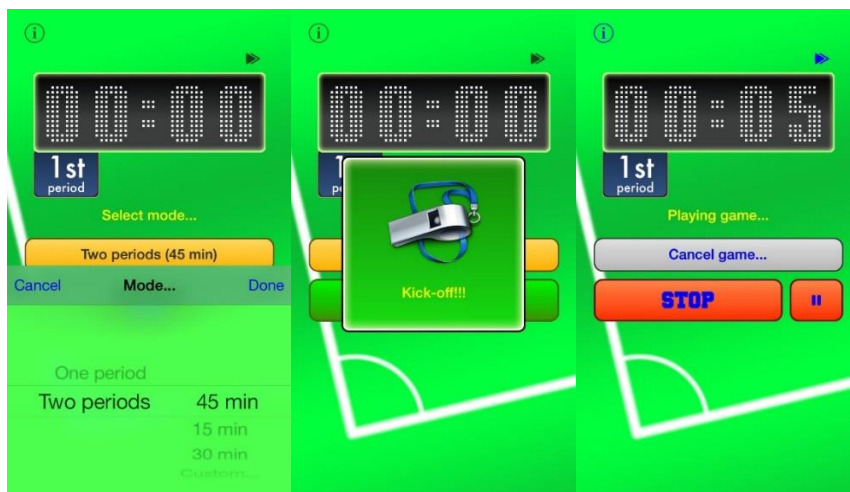


Ilustración 2.2 Capturas Crono Fútbol



Ilustración 2.3 Icono Goref

Goref (2016) es una aplicación para iPhone e iWatch para árbitros de futbol, que permite crear equipos, añadir equipos, saber el tiempo, goles, sustituciones y tarjetas.

La aplicación móvil permite la creación de distintos equipos, personalización de tiempo de partidos y personalización de equipos antes de iniciar el partido.

Tras la finalización del partido el teléfono puede mostrar el acta del partido con la información detallada de lo ocurrido a lo largo de este.

En las siguientes capturas podemos apreciar algunas de las pantallas de la aplicación móvil.

En la primera captura podemos ver la portada de la aplicación donde se pueden crear equipos, crear partidos o ver partidos.

En la segunda captura podemos ver como se configuran los partidos de la aplicación.

En la tercera de las imágenes, el usuario selecciona los jugadores de cada equipo creado

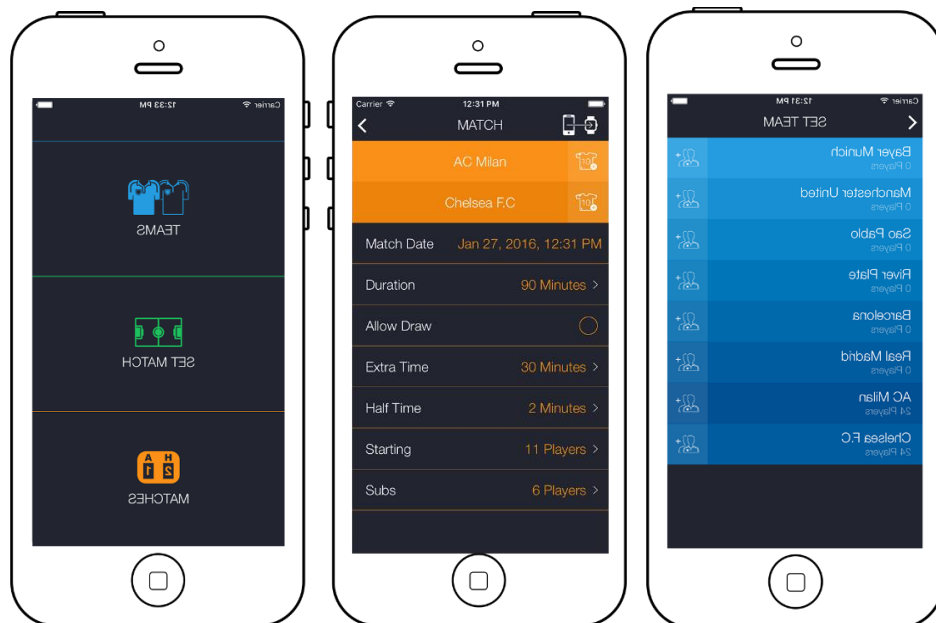


Ilustración 2.4 Capturas Goref móvil

La aplicación wear permite llevar el tiempo del partido con distintos cronómetros, añadir incidencias, ya sean goles sustituciones o tarjetas y almacenarlas para enviar tras terminar el partido.

En las siguientes capturas podemos ver las distintas pantallas de la aplicación wear.



Ilustración 2.5 Capturas Goref Wear

La mayor contra de esta aplicación es que al utilizar como dispositivo el reloj iWatch es necesario el uso de iPhone para utilizarla, y ambos dispositivos tienen un coste muy elevado.

2.2. Android

Cronómetro De Fútbol APK

El cronómetro De Fútbol APK (2016) es una aplicación para teléfono Android que permite saber el tiempo transcurrido durante los partidos y los goles de los equipos.

Esta aplicación está diseñada especialmente para llevar el conteo de goles y de tiempo de partidos amateur, es una aplicación cómoda de manejar, pero no precisa de muchas de las opciones necesarias para la gestión de un partido a nivel competitivo.

Como podemos apreciar en las capturas la aplicación, las principales funciones son, el manejo del cronómetro con los botones de iniciar pausar o terminar y la opción de añadir goles.

Una de las opciones más interesantes que presenta la aplicación es la posibilidad de bloquear los botones de la pantalla, gracias a esta opción se puede evitar pulsar botones por error durante el partido y así evitar preocuparte de que al tener el móvil en el bolsillo se pulse un botón por error.



Ilustración 2.6 Capturas Cronómetro De Fútbol APK

Soccer Referee Pro

Soccer Referee Pro (2018) es una aplicación para Android y Android Wear para árbitros de fútbol que permite gestionar el tiempo, goles, sustituciones y tarjetas a lo largo del partido.

La aplicación móvil permite la creación de distintos equipos, personalización de tiempo de partidos y personalización de los colores de cada equipo antes de iniciar el partido.

Tras la finalización del partido el teléfono puede mostrar el acta del partido con la información detallada de lo ocurrido a lo largo de este.

La aplicación dispone de integración con google calendar para programar distintos partidos en el calendario y avisar de la fecha de los partidos programados

En las siguientes capturas podemos apreciar algunas de las pantallas de las aplicaciones.

En la primera captura podemos ver la pantalla principal del reloj donde podemos controlar el tiempo del partido.

En la segunda de las pantallas se permite la gestión de las distintas incidencias del partido de forma rápida.

En la última pantalla podemos apreciarla vista del partido una vez terminado en el teléfono para ver los datos del partido en detalle.

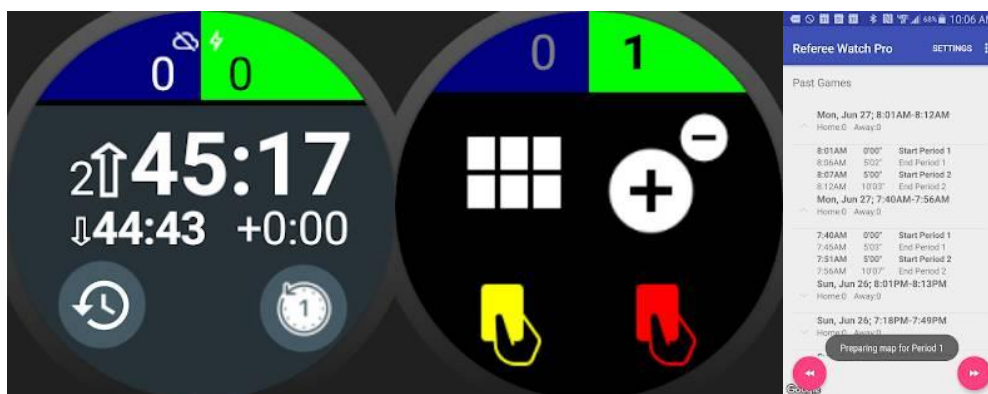


Ilustración 2.7 Capturas Soccer Referee Pro

Las desventajas de esta aplicación es que esta en inglés no dispone de español como idioma, y que la aplicación requiere de compra por 4,99€

Refsix

Refsix (2018) es una aplicación para Android y Android Wear para árbitros de fútbol, que permite crear equipos, añadir equipos, gestionar el tiempo, goles, sustituciones y tarjetas.

La aplicación móvil permite la creación de distintos equipos, personalización de tiempo de partidos y personalización de los equipos con anterioridad al partido.

Tras la finalización del partido el teléfono puede mostrar el acta del partido con la información detallada de lo ocurrido a lo largo de este.

La aplicación permite el seguimiento de la actividad deportiva del usuario, así como mapa de calor de la zona por donde se ha posicionado el usuario.

En las siguientes capturas podemos apreciar algunas de las pantallas de las aplicaciones.

En la primera captura podemos ver la pantalla principal del reloj y la pantalla de configuración del teléfono.

En la segunda de las pantallas se permite la gestión de las distintas incidencias del partido de forma rápida.

En la última pantalla podemos apreciarla vista del partido una vez terminado en el teléfono para ver los datos del partido en detalle.



Ilustración 2.8 Capturas Refsix

Las desventajas de esta aplicación es que requiere de un registro previo, para disponer de todas las funcionalidades se requiere de un pago mensual de 5€ y no dispone de español como idioma.

2.3. Comparativa

	Crono Fútbol	Goref	Cronómetro De Fútbol APK	Soccer Referee Pro	Refsix	Refwatch
SO	IOS	IOS	Android	Android	Android	Android
Idioma	Inglés	Español	Español	Inglés	Inglés	Español
Dispositivo	Teléfono	Teléfono/Reloj	Teléfono	Teléfono/Reloj	Teléfono/Reloj	Teléfono/Reloj
Goles	x	✓	✓	✓	✓	✓
Sustituciones	x	✓	x	✓	✓	✓
Tarjetas	✓	✓	x	✓	✓	✓
Vinculación con calendario	x	x	x	✓	x	Futuras versiones
Personalizar equipos	x	✓	x	x	✓	✓
Control del tiempo	✓	✓	✓	✓	✓	✓
Alerta de vibración	x	✓	x	x	x	✓

Tabla 1 Comparativa de aplicaciones

Como podemos apreciar en la tabla anterior, el sistema propuesto cumple con las funcionalidades del resto de aplicaciones, aglutinando las distintas funcionalidades de varios sistemas para conseguir ser una alternativa completa al resto de aplicaciones que se encuentran ahora mismo en el mercado.

Capítulo 3 Planificación y presupuesto

A continuación se hablará sobre la metodología, la planificación y la estimación de presupuesto del desarrollo de este proyecto.

3.1. Metodología

Debido a que durante el proyecto se añadirán funcionalidades en función del progreso del mismo, la metodología elegida será la una metodología incremental, ya que permitirá añadir nuevas funcionalidades en cada iteración.

El proyecto constará de 5 iteraciones, las cuales contarán de las siguientes fases:

Análisis: a lo largo de esta fase se realizará un estudio de los requisitos necesarios para un buen desarrollo de la aplicación.

Constará de distintos tipos de requisitos:

- Requisitos de usuario: acciones realizadas por los usuarios durante el uso de la aplicación.
- Requisitos funcionales: lo que el sistema puede hacer.
- Requisitos de información: datos utilizados durante la ejecución de la aplicación.

Diseño: durante esta fase se busca plantear el diseño de la aplicación tanto de forma estética, como funcional y de arquitectura.

Implementación: esta etapa será la fase en la que se desarrollará la aplicación.

Pruebas: una vez realizada la fase de implementación se realizarán pruebas de caja negra y de caja blanca sobre el código implementado para detectar fallos.

Documentación: En este apartado se redactará lo realizado a lo largo del resto de fases.

3.2 Estimación temporal

Como se ha mencionado en el apartado anterior constará de 5 iteraciones mediante metodología incremental, sin embargo el tiempo empleado en cada una de las iteraciones variará debido a la situación laboral debido a las prácticas de empresa.

Como podemos observar en las siguientes imágenes, durante las primeras fases se invertirá más tiempo en análisis y diseño, mientras que en las últimas fases del proyecto se dará más prioridad a la implementación pruebas y documentación.

En la siguiente imagen podemos observar la planificación temporal del proyecto dividida en 5 iteraciones.

Etapas proyecto	Días	Inicio	Final
Primera iteración	31.0	12-12-17	24-01-18
Analisis	9.0	12-12-17	22-12-17
Diseño	7.0	26-12-17	03-01-18
Implementación	7.0	04-01-18	12-01-18
Pruebas	3.0	13-01-18	17-01-18
Documentación	5.0	18-01-18	24-01-18
Segunda iteración	25.0	26-01-18	01-03-18
Analisis	8.0	26-01-18	06-02-18
Diseño	6.0	07-02-18	14-02-18
Implementación	4.0	15-02-18	20-02-18
Pruebas	2.0	21-02-18	22-02-18
Documentación	5.0	23-02-18	01-03-18
Tercera iteración	27.0	03-03-18	10-04-18
Analisis	7.0	03-03-18	13-03-18
Diseño	5.0	14-03-18	20-03-18
Implementación	8.0	21-03-18	30-03-18
Pruebas	3.0	31-03-18	04-04-18
Documentación	4.0	05-04-18	10-04-18
Cuarta iteración	24.0	12-04-18	15-05-18
Analisis	3.0	12-04-18	16-04-18
Diseño	2.0	17-04-18	18-04-18
Implementación	11.0	19-04-18	03-05-18
Pruebas	4.0	04-05-18	09-05-18
Documentación	4.0	10-05-18	15-05-18
Quinta iteración	26.0	17-05-18	21-06-18
Analisis	4.0	17-05-18	22-05-18
Diseño	2.0	23-05-18	24-05-18
Implementación	9.0	25-05-18	06-06-18
Pruebas	3.0	07-06-18	11-06-18
Documentación	8.0	12-06-18	21-06-18

133.0

Nombre del proyecto: Planificación Refwatch
Fecha: 12/12/2017

Ilustración 3.1 Planificación temporal

3.3 Estimaciones

Para realizar una estimación de las horas de elaboración del proyecto, se ha optado por utilizar dos tipos de modelos, el modelo de estimación por puntos de función, y el modelo COCOMO 2

3.3.1. Puntos de función

Este modelo busca medir el tamaño del software a desarrollar. Para lograr esto, se mide la funcionalidad que se pretende realizar sin tener en cuenta las tecnologías con las que se pretende llevar a cabo hasta que obtengamos los puntos de función.

Se realizarán los puntos de función de la aplicación móvil y la aplicación de Android Wear por separado.

Para evaluar ambos sistemas se evaluarán los siguientes parámetros:

- Entradas: datos introducidos por el usuario
- Salidas: datos mostrados por el sistema.
- Ficheros internos: base de datos interna del sistema.
- Ficheros externos: base de datos externa del sistema.
- Consultas externas: solicitud de datos a servidor externo.

Para definir la complejidad de cada parámetro utilizaremos la siguiente tabla:

Parámetros	Baja	Media	Alta
Entradas	3	4	6
Salidas	4	5	7
Ficheros internos	7	10	15
Ficheros externos	5	7	10
Consultas externas	3	7	6

Tabla 2 Complejidad según parámetros

Debido a que ambas aplicaciones están desarrolladas en java, se utilizará la correspondencia entre PF y líneas de código en java indicada en la imagen 3.2:

LOC por punto de función			
Lenguaje	LOC/FP	Lenguaje	LOC/FP
Ensamblador	320	Basic ANSI/Quick/Turbo	64
Macroensamblador	213	Java	53
C	150	Visual C++	34
Fortran	106	Foxpro 2,5	34
Cobol	106	Visual Basic	32
Pascal	91	Delphi	29
Cobol ANSI 85	91	C++	29
Basic	91	Visual Cobol	20
RPG	80	Clipper	19
PL/I	80	Power Builder	16
Ada	71	Hoja de Calculo	6

Tabla 3 LDC por PF

Aplicación móvil

A lo largo de la aplicación móvil se han encontrado los siguientes parámetros:

Parámetro	Componente	Complejidad
Entradas	Configuración de partido	Baja
	Configuración de equipos	Alta
Salidas	Ver partidos pendientes	Media
	Ver partidos terminados	Alta
	Notificación error	Baja
Ficheros internos	Base de datos app	Alta
Ficheros externos	-	
Consultas externas	Obtener partido	Alta
	Enviar partido	Alta

Tabla 4 Complejidad de componentes de aplicación móvil

Utilizando la tabla de complejidades dicha anteriormente calculamos los puntos de función sin ajustar correspondientes según los parámetros anteriores.

Parámetro	Complejidad	Total x Complejidad	Total
Entradas	Baja	1 x 3	9
	Media	0 x 4	
	Alta	1 x 6	
Salidas	Baja	1 x 4	16
	Media	1 x 5	
	Alta	1 x 7	
Ficheros internos	Baja	0 x 7	15
	Media	0 x 10	
	Alta	1 x 15	
Ficheros Externos	Baja	0 x 5	0
	Media	0 x 7	
	Alta	0 x 10	
Consultas externas	Baja	0 x 3	20
	Media	0 x 7	
	Alta	2 x 10	
			60

Tabla 5 Tabla de puntos de función sin ajustar de aplicación móvil

Una vez obtenidos los puntos de función de la aplicación calculamos el valor de los distintos factores de complejidad, para ello, fijarse en el anexo I sobre atributos del ajuste de los puntos de función.

Factor de complejidad	Valor elegido
Comunicación de datos	4
Funciones distribuidas	1
Prestaciones	2
Gran uso de la configuración	2
Velocidad de transacciones	1
Entrada de datos en línea	0
Diseño para la eficiencia del usuario final	1
Actualización de datos en línea	3
Complejidad del proceso lógico interno de la aplicación	2
Reusabilidad del código	5
Facilidad de instalación	1
Facilidad de operación	0
Localizaciones múltiples	0
Facilidad de cambios	3
25	

Tabla 6 Coeficientes para el factor de ajuste de aplicación móvil

Una vez obtenidos los coeficientes de ajuste calculamos el factor de ajuste con la siguiente fórmula:

$$FA = (0.01 \times \text{total coeficientes}) + 0.65$$

$$FA = (0.01 \times 25) + 0.65 = 0.9$$

Una vez calculado el factor de ajuste debemos multiplicarlo por los puntos de función para obtener los puntos de función ajustados:

$$PFA = PF \times FA$$

$$PFA = 60 \times 0.9 = 54$$

Tras obtener la correspondencia de los puntos de función a líneas de código calcularemos las líneas de código:

$$LDC = 53 \times 54 = 2862$$

Aplicación wear

A lo largo de la aplicación de Android Wear se han encontrado los siguientes parámetros:

Parámetro	Componente	Complejidad
Entradas	Añadir goles	Alta
	Añadir sustituciones	Alta
	Añadir tarjetas	Alta
Salidas	Ver cronómetros	Media
	Visualizar goles	Baja
	Visualizar equipos	Baja
	Visualizar jugadores	Alta
	Visualizar tarjetas	Media
Ficheros internos	Base de datos app	Alta
Ficheros Externos	-	
Consultas externas	Obtener partido	Alta
	Enviar partido	Alta

Tabla 7 Complejidad de componentes de aplicación Wear

Utilizando la tabla de complejidades dicha anteriormente calculamos los puntos de función sin ajustar correspondientes según los parámetros anteriores.

Parámetro	Complejidad	Total x Complejidad	Total
Entradas	Baja	0 x 3	18
	Media	0 x 4	
	Alta	3 x 6	
Salidas	Baja	2 x 4	25
	Media	2 x 5	
	Alta	1 x 7	
Ficheros internos	Baja	0 x 7	15
	Media	0 x 10	
	Alta	1 x 15	
Ficheros Externos	Baja	0 x 5	0
	Media	0 x 7	
	Alta	0 x 10	
Consultas externas	Baja	0 x 3	20
	Media	0 x 7	
	Alta	2 x 10	
			78

Tabla 8 Coeficientes para el factor de ajuste de aplicación Wear

Una vez obtenidos los puntos de función de la aplicación calculamos el valor de los distintos factores de complejidad, para ello, fijarse en el anexo I sobre atributos del ajuste de los puntos de función.

Factor de complejidad	Valor elegido
Comunicación de datos	4
Funciones distribuidas	1

Prestaciones	2
Gran uso de la configuración	2
Velocidad de transacciones	1
Entrada de datos en línea	0
Diseño para la eficiencia del usuario final	1
Actualización de datos en línea	3
Complejidad del proceso lógico interno de la aplicación	2
Reusabilidad del código	5
Facilidad de instalación	1
Facilidad de operación	0
Localizaciones múltiples	0
Facilidad de cambios	3
	25

Tabla 9 Coeficientes para el factor de ajuste de aplicación Wear

Una vez obtenidos los coeficientes de ajuste calculamos el factor de ajuste con la siguiente fórmula:

$$FA = (0.01 \times \text{total coeficientes}) + 0.65$$

$$FA = (0.01 \times 25) + 0.65 = 0.9$$

Una vez calculado el factor de ajuste debemos multiplicarlo por los puntos de función para obtener los puntos de función ajustados:

$$PFA = PF \times FA$$

$$PFA = 78 \times 0.9 = 70.2$$

Tras obtener la correspondencia de los puntos de función a líneas de código calcularemos las líneas de código:

$$LDC = 53 \times 70.2 = 3720$$

En total de los dos sistemas tendríamos:

$$LDC = 3720 + 2862 = 6582 \approx 6.5KLDC$$

3.3.2. COCOMO 2

COCOMO es un tipo de estimación basada en la medición del esfuerzo y el tiempo. Para utilizar este método necesitaremos usar las líneas de código, pero variara en función del procedimiento utilizado, COCOMO dispone de 3 modelos distintos en función del sistema requerido:

MODELO	A	B	C	d
Orgánico	3.20	1.05	2.50	0.38
Semi-libre	3.00	1.12	2.50	0.35
Empotrado	2.80	1.20	2.50	0.32

Tabla 10 Modelos de COCOMO

Debido a que es un proyecto desarrollado por un equipo pequeño y con poca rigidez se utilizará un modelo orgánico.

FACTORES	Valor de los factores					
	Muy bajo	Bajo	Medio	Alto	Muy alto	Extra
Fiabilidad requerida	0,75	0,88	1,00	1,15	1,4	
Tamaño de la base de datos		0,94	1,00	1,08	1,16	
Complejidad del software	0,70	0,85	1,00	1,15	1,30	1,65
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria			1,00	1,06	1,21	1,56
Volatilidad del hardware		0,87	1,00	1,15	1,30	
Restricciones de tiempo de respuesta		0,87	1,00	1,07		
Calidad de los analistas	1,46	1,19	1,00	0,86	0,71	
Experiencia con el tipo de aplicación	1,29	1,13	1,00	0,91	0,82	
Experiencia con el hardware	1,21	1,10	1,00	0,90		
Exp. con el lenguaje de programación.	1,14	1,07	1,00	0,95		
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Técnicas modernas de programación	1,24	1,10	1,00	0,91	0,82	
Empleo de herramientas	1,24	1,10	1,00	0,91	0,83	
Restricciones a la duración del proyec.	1,23	1,08	1,00	1,04	1,10	

Tabla 11 Factores de ajuste de COCOMO

Hemos asignado los siguientes valores y hemos calculado el total realizando el producto de los valores:

Fiabilidad requerida	Alto	1.15
Tamaño de la base de datos	Bajo	0.94
Complejidad del software	Bajo	0.85
Restricciones de tiempo de ejecución	Alto	1.11
Restricciones de memoria	Medio	1.21
Volatilidad del hardware	Bajo	0.87
Restricciones de tiempo de respuesta	Alto	1.07
Calidad de los analistas	Alto	0.86
Experiencia con el tipo de aplicación	Medio	1.00
Experiencia con el hardware	Alto	0.90
Experiencia con el lenguaje de programación	Alto	0.95
Calidad de los programadores	Medio	1.00
Técnicas modernas de programación	Medio	1.00
Empleo de herramientas	Medio	1.00
Restricciones a la duración del proyecto	Bajo	1.08
	Total	0,91231396

Tabla 12 Elección de factores de ajuste de COCOMO

Para calcular el esfuerzo utilizaremos la siguiente formula:

$$E = a \cdot (KLDC)^b \times \text{factor de ajuste}$$

$$\text{Esfuerzo} = 2.4 \times 6.5^{1.05} \times 0.91231 = 15.59 \text{ personas / mes}$$

Para calcular el tiempo de desarrollo:

$$\text{Tiempo de desarrollo} = c \times E^d$$

$$\text{Tiempo de desarrollo} = 2.5 \times 15.59^{0.38} = 7.1 \text{ meses}$$

El número medio de personas para realizar el proyecto con los resultados obtenidos es:

$$N^{\circ} \text{ medio de personas} = 15.59 \div 7.1 = 2.2 \approx 3 \text{ personas}$$

Tras ver los resultados de la estimación podemos observar cómo se necesitaría de 3 personas trabajando durante 7.1 meses para realizar el proyecto, sin embargo como este proyecto será realizado por una única persona se estima que se realizará en cerca de 16 meses.

3.4. Presupuesto

En este apartado se representará el coste del proyecto inicialmente y el coste del proyecto real una vez finalizado el mismo, estimando las distintas partes del mismo: presupuesto hardware, presupuesto software y presupuesto de personal. Una vez

3.4.1. Presupuesto inicial

Se realizará el presupuesto para 6 meses según la estimación inicial realizada

Presupuesto hardware

En la siguiente tabla se muestran todos los componentes hardware utilizados a lo largo del proyecto. Para calcular el coste de los distintos componentes, calcularemos la vida útil del componente en función del tiempo que lo hemos utilizado obteniendo de esta forma el coste real del componente para nuestro proyecto.

Componente	Vida útil	% de uso	Coste total	Coste real
Ordenador sobremesa	4 años	12.5%	900 €	112.5 €
Teléfono de pruebas	2 años	25%	142 €	35.5 €
Reloj de pruebas	2 años	25%	110 €	27.5 €
Internet	1 mes	50%	30 € x5 meses	90 €

Tabla 13 Presupuesto hardware

Sumando el total de costes tenemos un total de 265.5 € de hardware.

Presupuesto software

En la siguiente tabla se muestran todos los componentes software utilizados a lo largo del proyecto. Para calcular el coste de los distintos componentes, calcularemos la vida útil del componente en función del tiempo que lo hemos utilizado obteniendo de esta forma el coste real del componente para nuestro proyecto.

Componente	Vida útil	% de uso	Coste total	Coste real
Windows 10	4 años	12.5%	145 €	18.13 €
Android Studio			Gratis	0 €
Word			Licencia de estudiante	0 €
Tomsplanner			Gratis	0 €
Draw.io			Gratis	0 €
Wear OS			Gratis	0 €

Tabla 14 Presupuesto software

Realizando la suma total de los costes obtenemos un coste de 18.13 € de software.

Presupuesto personal

Teniendo en cuenta los planificados en la estimación temporal de 133 días, lo que equivaldría a 665 horas, teniendo en cuenta 5 horas al día de trabajo, se realizará el presupuesto asumiendo el coste de un analista y de un programador, teniendo en cuenta que el analista realizará las tareas de análisis, diseño y documentación y un programador realizará las fases de implementación y pruebas.

Personal	Coste x hora	Horas	Total
Analista	14 €/hora	395	5530 €
Programador	10 €/hora	270	2700 €

Tabla 15 Presupuesto de personal

Realizando la suma total de costes de personal obtenemos un coste de 8230 € en personal.

Presupuesto inicial total

Sumando los importes de los anteriores presupuestos obtenemos el siguiente coste total:

Presupuesto	Coste
Hardware	265.5 €
Software	18.13 €
Personal	8230 €
Total	8513.63€

Tabla 16 Presupuesto total

3.4.2. Presupuesto real

Se realizará el presupuesto para 6 meses y medio que fueron necesarios para el desarrollo final del proyecto.

Presupuesto hardware

En la siguiente tabla se muestran todos los componentes hardware utilizados a lo largo del proyecto. Para calcular el coste de los distintos componentes, calcularemos la vida útil del componente en función del tiempo que lo hemos utilizado obteniendo de esta forma el coste real del componente para nuestro proyecto.

Componente	Vida útil	% de uso	Coste total	Coste real
Ordenador sobremesa	4 años	13.55%	900 €	121.95€
Teléfono de pruebas	2 años	27%	142 €	38.34 €
Reloj de pruebas	2 años	27%	110 €	29.7 €
Internet	1 mes	46.4%	30 € x7 meses	97.44 €

Tabla 17 Presupuesto hardware real

Sumando el total de costes tenemos un total de 287.43 € de hardware.

Presupuesto software

En la siguiente tabla se muestran todos los componentes software utilizados a lo largo del proyecto. Para calcular el coste de los distintos componentes, calcularemos la vida útil del componente en función del tiempo que lo hemos utilizado obteniendo de esta forma el coste real del componente para nuestro proyecto.

Componente	Vida útil	% de uso	Coste total	Coste real
Windows 10	4 años	13.55%	145 €	20.87 €
Android Studio			Gratis	0 €
Word			Licencia de estudiante	0 €
Tomsplanner			Gratis	0 €
Draw.io			Gratis	0 €
Wear OS			Gratis	0 €

Tabla 18 Presupuesto software real

Realizando la suma total de los costes obtenemos un coste de 20.87 € de software.

Presupuesto personal

Teniendo en cuenta que se ha alargado el proyecto medio mes debido a problemas durante el desarrollo y a añadir nuevas funcionalidades el proyecto durará 6 meses y medio ampliándose de esta manera el tiempo de trabajo del personal.

Personal	Coste x hora	Horas	Total
Analista	14 €/hora	445	6230€
Programador	10 €/hora	320	3200 €

Tabla 19 Presupuesto de personal real

Realizando la suma total de costes de personal obtenemos un coste de 9430 € en personal.

Presupuesto inicial total

Sumando los importes de los anteriores presupuestos obtenemos el siguiente coste total:

Presupuesto	Coste
Hardware	287.43 €
Software	20.87 €
Personal	9430 €
Total	9738.3€

Tabla 20 Presupuesto total real

3.5. Conclusiones

El presupuesto real se ha desviado en más de 1200 € respecto al presupuesto inicial, esto se debe al aumento de funcionalidades añadidas durante el desarrollo que han supuesto mayor tiempo de lo esperado, dichas funcionalidades han supuesto una gran cantidad de modificaciones en varias de las etapas de desarrollo

Capítulo 4 Análisis

El proyecto se compone de dos aplicaciones, una aplicación Android para dispositivos móviles y una aplicación Android Wear para relojes. Al ser dos sistemas distintos, las especificaciones de estos estarán separadas cuando sea requerido.

En este capítulo se muestra el análisis previo a la implementación de ambos sistemas que se van a desarrollar.

4.1. Requisitos de usuario

En este apartado se muestra la interacción entre los actores y el sistema.

4.1.1. Descripción de los actores

Durante el análisis se han distinguido distintos actores en función del sistema con el que interactúan.

Aplicación Móvil

ACTOR	DESCRIPCIÓN
Usuario	Interactúa con la aplicación móvil tras abrir la aplicación.
Bluetooth	Su función es recibir y enviar partidos.

Tabla 21 Actores aplicación móvil

Aplicación Wear

ACTOR	DESCRIPCIÓN
Usuario	Interactúa con la aplicación wear tras iniciar el partido
Reloj	Su función es generar eventos de manera periódica.
Bluetooth	Su función es recibir y enviar partidos.

Tabla 22 Actores aplicación Wear

4.1.2. Listado de requisitos de usuario

En estas tablas se muestran los listados de requisitos de usuario para los sistemas de aplicación móvil y wear:

Aplicación Móvil

REQUISITO DE USUARIO	DESCRIPCIÓN
RU-01	El usuario podrá configurar un nuevo partido.
RU-02	El usuario podrá configurar un nuevo equipo.
RU-03	El usuario podrá guardar un nuevo partido.
RU-04	El usuario podrá ver los partidos pendientes.
RU-05	El usuario podrá ver un partido pendiente.
RU-06	El usuario podrá borrar un partido.
RU-07	El usuario podrá empezar un partido.
RU-08	El usuario podrá ver los partidos finalizados.
RU-09	El usuario podrá ver un partido finalizado.
RU-10	El usuario podrá borrar un partido finalizado.
RU-11	El bluetooth podrá recibir un partido terminado.
RU-12	El bluetooth podrá enviar un partido pendiente.

Tabla 23 Requisitos de usuario aplicación móvil

Aplicación Wear

REQUISITO DE USUARIO	DESCRIPCIÓN
RU-13	El usuario podrá iniciar o pausar el cronómetro.
RU-14	El usuario podrá cambiar el periodo del partido.
RU-15	El usuario podrá terminar el partido.
RU-16	El usuario podrá añadir goles.
RU-17	El usuario podrá añadir sustituciones.
RU-18	El usuario podrá añadir tarjetas.
RU-19	El reloj podrá actualizar los cronómetros.
RU-20	El reloj podrá finalizar los cronómetros.
RU-21	El bluetooth podrá recibir un partido.

Tabla 24 Requisitos de usuario aplicación Wear

4.1.3 Diagramas de casos de uso

Aplicación Móvil

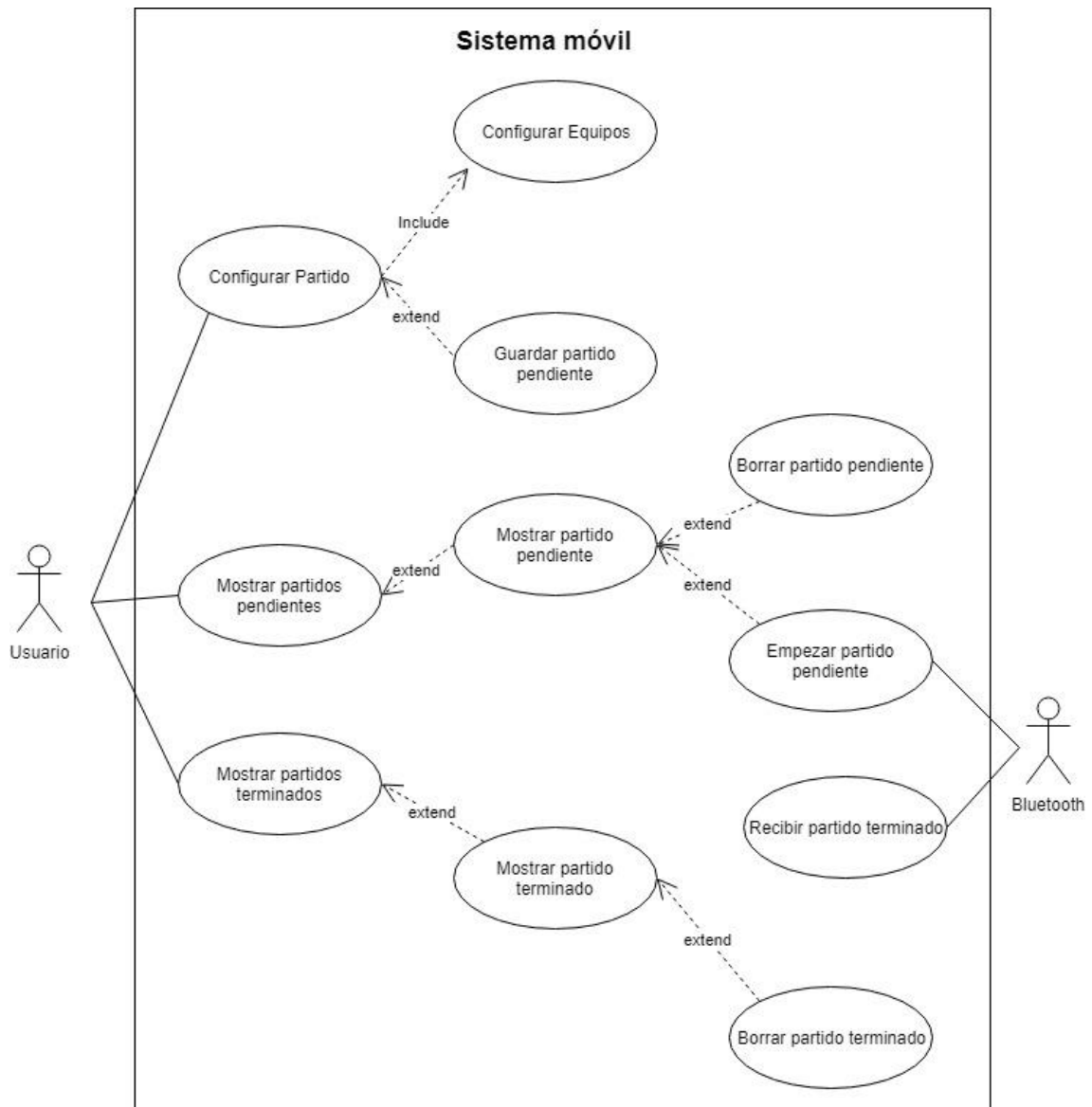


Ilustración 4.1 Diagrama de casos de uso de aplicación móvil

Aplicación Wear

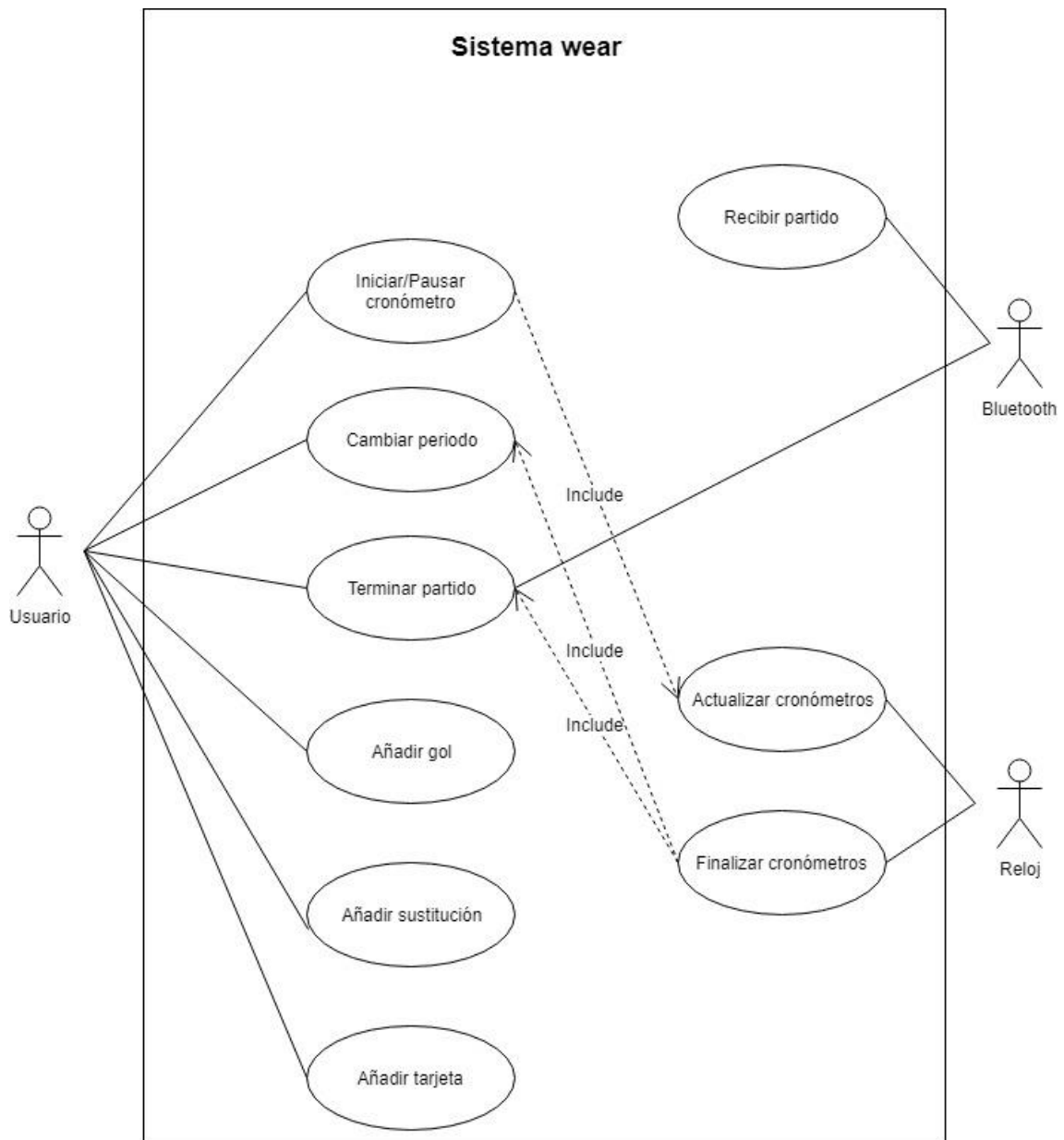


Ilustración 4.2 Diagrama de casos de uso de aplicación Wear

4.1.4. Especificación de casos de uso

En este apartado se mostrarán los casos de uso más representativos de la aplicación móvil y de la aplicación de Android Wear, el resto de los casos de uso se incluirán en un anexo para facilitar la lectura del documento.

Aplicación móvil

CU-02	Configurar equipos	
Versión	1.0	
Requisitos asociados	CU-01	
Actores	Usuario	
Descripción	El usuario configura los jugadores de un equipo.	
Precondiciones	Se deben de haber elegido previamente las opciones del partido.	
Secuencia normal	Paso	Acción
	P1	El usuario solicita configurar los jugadores de un equipo.
	P2	El sistema muestra un listado con todos los jugadores.
	P3	El usuario elige jugador a jugador los titulares.
	P4	El sistema marca como seleccionados los jugadores del equipo cambiando el color de la selección.
	P5	El sistema actualiza el contador de jugadores elegidos.
	P6	El usuario solicita seleccionar substitutos.
	P7	El sistema muestra un listado con todos los jugadores no seleccionados como titulares.
	P8	El usuario elige jugador a jugador los substitutos.
	P9	El sistema marca como seleccionados los jugadores del equipo cambiando el color de la selección.
	P10	El sistema actualiza el contador de jugadores elegidos.
	P11	El usuario solicita guardar la configuración del equipo.
P12	El caso de uso finaliza correctamente.	
Postcondiciones	La configuración del equipo queda guardada correctamente.	
Excepciones	Paso	Acción
	P3	Si al elegir un jugador este ya estaba seleccionado el jugador se desmarca.
	P3	Si al elegir un jugador ya se ha alcanzado el número máximo de jugadores no se podrá añadir dicho jugador al menos que se desmarque otro jugador previamente.
	P6	Si se seleccionan menos de 7 jugadores se mostrará una notificación indicando que se necesitan al menos 7 jugadores
	P8	Si al elegir un jugador este ya estaba seleccionado el jugador se desmarca.
	P8	Si al elegir un jugador ya se ha alcanzado el número máximo de jugadores no se podrá añadir dicho jugador al menos que se desmarque otro jugador previamente.
Frecuencia	Diaria	
Importancia	Alta. Función necesaria para las realizar sustituciones durante el partido.	

Comentarios	Tras completarse el caso de uso el botón de configuración del equipo correspondiente cambiará de color para indicar que se ha completado su configuración
-------------	---

Tabla 25 CU-02 Configurar equipos

CU-03	Guardar partido pendiente	
Versión	1.0	
Requisitos asociados	CU-01	
Actores	Usuario	
Descripción	Guarda el partido en la base de datos.	
Precondiciones	Datos correctamente validados.	
Secuencia normal	Paso	Acción
	P1	El usuario solicita guardar el partido
	P2	El sistema comprueba que todos los datos introducidos son válidos y que los equipos están configurados correctamente.
	P3	El sistema guarda el partido en la base de datos local.
	P4	El sistema notifica que el partido se ha guardado correctamente.
	P5	El sistema vuelve a la pantalla principal de la aplicación.
Postcondiciones	El partido se ha guardado correctamente	
Excepciones	Paso	Acción
	P2	El sistema muestra una notificación indicando que falta por introducir datos para guardar el partido
	P3	El sistema muestra una notificación indicando que el partido no se ha podido guardar en la base de datos correctamente.
Frecuencia	Diaria	
Importancia	Alta. Función necesaria para crear partidos.	
Comentarios		

Tabla 26 CU-03 Guardar partido pendiente

CU-04	Mostrar partidos pendientes	
Versión	1.0	
Requisitos asociados	CU-05	
Actores	Usuario	
Descripción	Muestra un listado de los partidos pendientes	
Precondiciones	-	
Secuencia normal	Paso	Acción
	P1	El usuario solicita mostrar el listado de partidos pendientes.
	P2	El sistema obtiene el listado de partidos pendientes de la base de datos local.
	P3	El sistema muestra el listado de los partidos pendientes.
	P4	El caso de uso finaliza correctamente.
Postcondiciones	Se muestran los partidos pendientes.	
Excepciones	Paso	Acción
	P2	Si no existen partidos en la base de datos no se mostrará nada.
Frecuencia	Diaria	
Importancia	Alta. Función necesaria para iniciar un partido	
Comentarios		

Tabla 27 CU-04 Mostrar partidos pendientes

CU-07	Empezar partido pendiente	
Versión	1.0	
Requisitos asociados	CU-05	
Actores	Usuario, Bluetooth	
Descripción	Manda los datos del partido mediante bluetooth.	
Precondiciones	Debe existir un partido.	
Secuencia normal	Paso	Acción
	P1	El usuario solicita enviar un partido.
	P2	El sistema solicita comprueba la conexión bluetooth.
	P3	El sistema manda el partido por bluetooth.
	P4	El sistema muestra una notificación de partido enviado correctamente.
	P5	El caso de uso finaliza correctamente.
Postcondiciones	El partido ha sido enviado.	
Excepciones	Paso	Acción
	P2	Si el sistema no tiene la conexión bluetooth habilitada se muestra un mensaje de error.
	P3	Si no se puede mandar el mensaje se muestra una notificación del error.
Frecuencia	Diaria	
Importancia	Alta. Es necesario realizar envíos de partidos al reloj.	
Comentarios	Para conectarse con el teléfono deberá estar el reloj sincronizado mediante Wear OS al Android wear para hacer uso de las API's del mismo.	

Tabla 28 CU-07 Empezar partido pendiente

Aplicación wear

CU-13	Terminar partido	
Versión	1.0	
Requisitos asociados	CU-20	
Actores	Usuario, Bluetooth, Reloj	
Descripción	Manda el partido por bluetooth	
Precondiciones	El partido debe de haber terminado.	
Secuencia normal	Paso	Acción
	P1	El usuario solicita terminar el partido.
	P2	El sistema comprueba que los datos del partido son correctos.
	P3	El sistema comprueba que el partido haya terminado previamente.
	P4	El sistema comprueba la conexión bluetooth.
	P5	El sistema envía el partido por bluetooth.
	P6	El sistema notifica que el partido se ha enviado con éxito.
	P7	El sistema cierra la aplicación wear.
P8	El caso de uso finaliza correctamente.	
Postcondiciones	El partido ha sido enviado.	
Excepciones	Paso	Acción
	P2	Si los datos son incorrectos el sistema mostrara una notificación con el error correspondiente.
	P3	Si el partido no ha terminado no se continuará con el envío del partido.
	P4	Si no se puede enviar el partido por bluetooth el sistema mostrara una notificación con el error correspondiente.
Frecuencia	Diaria	
Importancia	Alta. Es necesario el envío de partidos terminados al reloj	
Comentarios	Para conectarse con el teléfono deberá estar el reloj sincronizado mediante Wear OS al Android wear para hacer uso de las API's del mismo.	

Tabla 29 CU-13 Terminar partido

CU-14	Iniciar/pausar cronómetro	
Versión	1.0	
Requisitos asociados	CU-19	
Actores	Usuario, Reloj	
Descripción	Inicia o pausa el cronómetro.	
Precondiciones	El partido debe de estar inicializado.	
Secuencia normal	Paso	Acción
	P1	El usuario solicita iniciar/pausar cronometro
	P2	El sistema comprueba si el cronometro principal esta iniciado, en pausa o es su primer inicio.
	P3.1	Si el cronómetro esta pausado el sistema lo inicia y pausa el cronómetro de pausa.
	P3.2	Si el cronómetro esta iniciado el sistema lo pausa e inicia el cronómetro de pausa.
	P3.3	Si era el primer inicio del cronometro principal inicia el cronometro total
	P4	El caso de uso termina correctamente.
Postcondiciones	El cronometro se activa/desactiva.	
Excepciones	Paso	Acción
	P1	Si el cronómetro principal está a 0 el sistema no hace nada porque el periodo ya ha terminado
Frecuencia	Constante.	
Importancia	Vital. Función principal de la aplicación para controlar los cronómetros	
Comentarios		

Tabla 30 CU-13 Iniciar/pausar cronómetro

CU-15	Cambiar periodo	
Versión	1.0	
Requisitos asociados	CU-20	
Actores	Usuario, Reloj	
Descripción	Cambia el periodo del partido	
Precondiciones		
Secuencia normal	Paso	Acción
	P1	El usuario solicita cambiar de periodo.
	P2	El sistema comprueba el tiempo del cronómetro principal.
	P3	El sistema comprueba que periodo es y si puede haber prorroga.
	P4	El sistema inicializa el cronómetro principal con el tiempo correspondiente al siguiente periodo.
	P5	El sistema modifica el cronómetro total, eliminando el tiempo de pausa.
	P6	El sistema inicializa el cronómetro de pausa a 0 para el nuevo periodo.
	P7	El sistema modifica el texto del periodo actual indicando el nuevo periodo.
	P8	El caso de uso finaliza correctamente.
Postcondiciones	El periodo ha sido cambiado.	
Excepciones	Paso	Acción
	P2	Si el cronometro principal no ha llegado a 0, el sistema mostrara una notificación de que el periodo aún no ha terminado.
	P3	Si el periodo actual es la segunda parte y no hay prorroga, el partido ha terminado y no se cambiará de periodo
	P3	Si el periodo actual es la segunda parte y hay prorroga, el partido terminará si el resultado es distinto de empate.
Frecuencia	Diaria	
Importancia	Vital. Función principal de la aplicación para controlar los cronómetros	
Comentarios		

Tabla 31 CU-15 Cambiar periodo

CU-16	Añadir gol	
Versión	1.0	
Requisitos asociados		
Actores	Usuario	
Descripción	Añade una incidencia de tipo gol	
Precondiciones	El periodo no puede haber terminado	
Secuencia normal	Paso	Acción
	P1	El usuario solicita añadir un gol.
	P2	El sistema muestra un listado con los jugadores.
	P3	El usuario elige la pestaña del equipo que metió el gol.
	P4	El usuario selecciona el tipo de gol.
	P5	El usuario selecciona el jugador que marcó el gol.
	P6	El sistema añade la incidencia.
	P7	El caso de uso termina correctamente.
Postcondiciones	Se ha añadido una nueva incidencia de tipo gol.	
Excepciones		
Frecuencia	Diaria	
Importancia	Alta. Es importante la gestión de las incidencias.	
Comentarios	Un gol puede ser de tipo: Gol, Autogol, Penal.	

Tabla 32 CU-16 Añadir gol

4.2. Requisitos de información

Tras el análisis realizado en los dos puntos anteriores, obtenemos los siguientes requisitos de información:

- RI-01: Se guardará la configuración de los partidos creados por el usuario, guardando los tiempos del partido así como las preferencias elegidas por él usuario.
- RI-02: El sistema guardará para los equipos, los jugadores titulares, los jugadores sustitutos y el nombre del equipo
- RI-03: El sistema guardara de las incidencias, el tiempo en el que ocurrió la incidencia y los implicados en ella.
- RI-04: El sistema almacenará los resultados del partido tras recibirles, incluyendo los equipos, incidencias y la configuración realizada para el partido.

4.2.1. Modelo de datos conceptual

Para modelar la información se ha optado por su representación mediante un diagrama Entidad-Relación. Los atributos del diagrama serán detallados en el diccionario de datos.

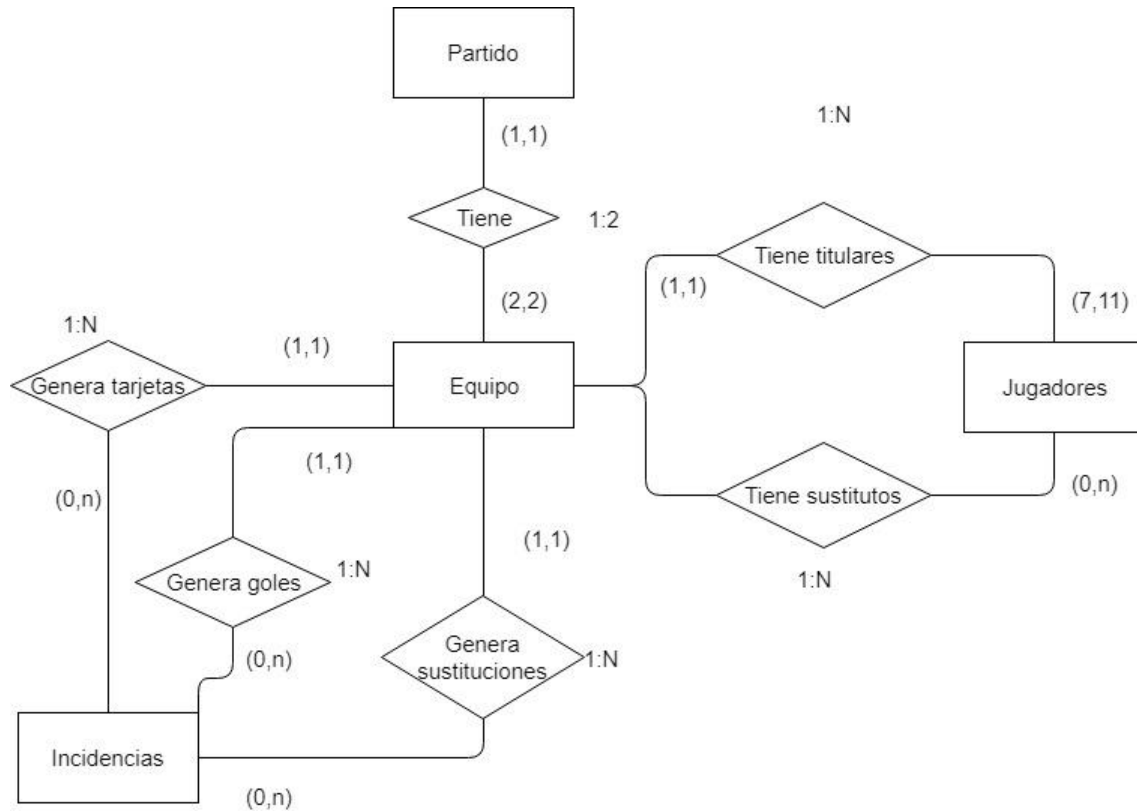


Ilustración 4.3 Modelo Entidad-Relación

4.2.2. Diccionario de datos

En el diccionario de datos se describirán las entidades almacenadas en la base de datos, su significado y formato.

A continuación se detallan las entidades utilizadas:

Partido: almacena la configuración del partido elegida por el usuario.

Atributos	Descripción	Tipo de dato	Dominio	Otro
id	Identificador único del partido	Número	Numero entero de hasta 5 cifras	Auto incremental
Gol local	Contador de goles del equipo local	Número	Numero entero de 2 cifras	Obligatorio
Gol visitante	Contador de goles del equipo visitante	Número	Numero entero de 2 cifras	Obligatoria
Fecha	Fecha del partido	Date	Día/mes/año	Obligatorio

Tabla 33 Entidad Partido

Equipo: almacena la información del equipo y la configuración del usuario.

Atributos	Descripción	Tipo de dato	Dominio	Otro
id	Identificador único del equipo	Número	Numero entero de hasta 5 cifras	Auto incremental
Nombre	Nombre del equipo	Texto	Cadena de caracteres con longitud máxima de 40	Obligatorio

Tabla 34 Entidad Equipo

Jugador: almacena la información del jugador.

Atributos	Descripción	Tipo de dato	Dominio	Otro
id	Identificador único del jugador	Número	Numero entero de hasta 5 cifras	Auto incremental
Nombre	Nombre del jugador, actualmente representado por el número de dorsal del jugador	Texto	Cadena de caracteres con longitud máxima de 40	Obligatorio
Tarjetas	Número de tarjetas del jugador	Número	Numero entero de hasta 5 cifras	Obligatorio

Tabla 35 Entidad Jugador

Incidencia: almacena la información del jugador.

Atributos	Descripción	Tipo de dato	Dominio	Otro
id	Identificador único de la incidencia	Número	Numero entero de hasta 5 cifras	Auto incremental
Tiempo	Tiempo en el que se da lugar la incidencia	Texto	Cadena de caracteres con longitud máxima de 40	Obligatorio
Jugador	Nombre del jugador que realiza la incidencia	Texto	Cadena de caracteres con longitud máxima de 40	Obligatorio
Comentarios	Comentarios realizados durante la incidencia	Texto	Cadena de caracteres con longitud máxima de 200	Nullable
Tipo	Tipo de incidencia cometida, puede ser gol, sustitución o tarjeta	Texto	Cadena de caracteres con longitud máxima de 40	Obligatorio

Tabla 36 Entidad Incidencia

Capítulo 5 Diseño

En este capítulo se expone el diseño que se tendrá en cuenta a la hora de desarrollar el sistema. Se tratarán las arquitecturas lógicas y físicas, el diagrama de clases, diagramas de secuencias representativos del sistema, el modelo lógico de datos y los diseños de algunas de las pantallas de interfaz de usuario del sistema.

5.1. Arquitectura lógica

En este apartado se representará la arquitectura lógica del sistema, describiendo los distintos componentes del sistema y sus relaciones.

Debido al uso de la misma arquitectura en ambos sistemas el siguiente diagrama representa a ambos sistemas.

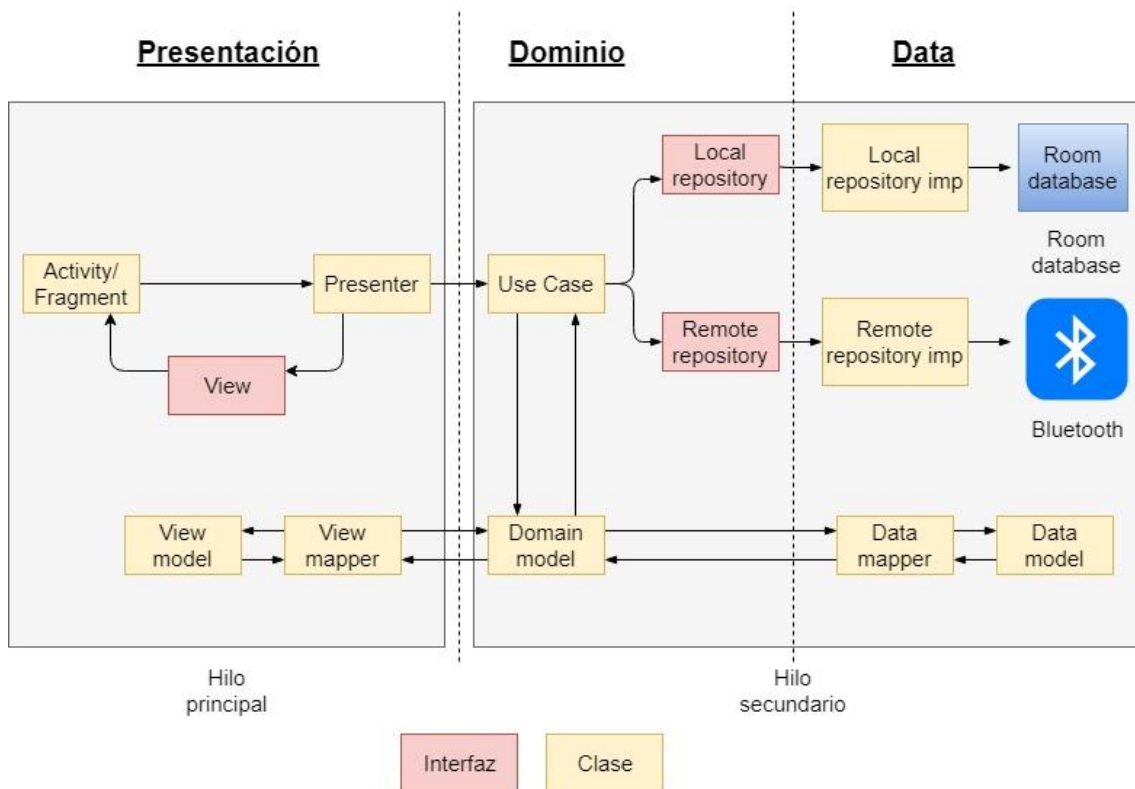


Ilustración 5.1 Arquitectura lógica

El diagrama anterior representa la arquitectura clean utilizada para nuestros sistemas.

Esta arquitectura surgió como respuesta a la pregunta:

¿Cómo separar la lógica de interfaz de usuario y la lógica de negocio?

Para lograr esto se utiliza un modelo de 3 capas, la capa de presentación, la capa de dominio y la capa de data mediante las cuales conseguiremos:

- Independencia de la interfaz de usuario: podremos modificar cualquier elemento de la interfaz de usuario sin necesidad de realizar cambios en la lógica de negocio.
- Independencia de la base de datos: podremos intercambiar la fuente de obtención de datos sin afectar al resto de la aplicación.
- Facilidad de realizar testeos: debido a la división en capas, la capa de dominio puede ser testeada sin ningún otro elemento de interfaz de usuario o de acceso a datos.
- Aumento de seguridad de la aplicación: debido a las dependencias entre las distintas capas los casos de uso no tendrán acceso a la información del exterior, como podemos observar en la siguiente imagen

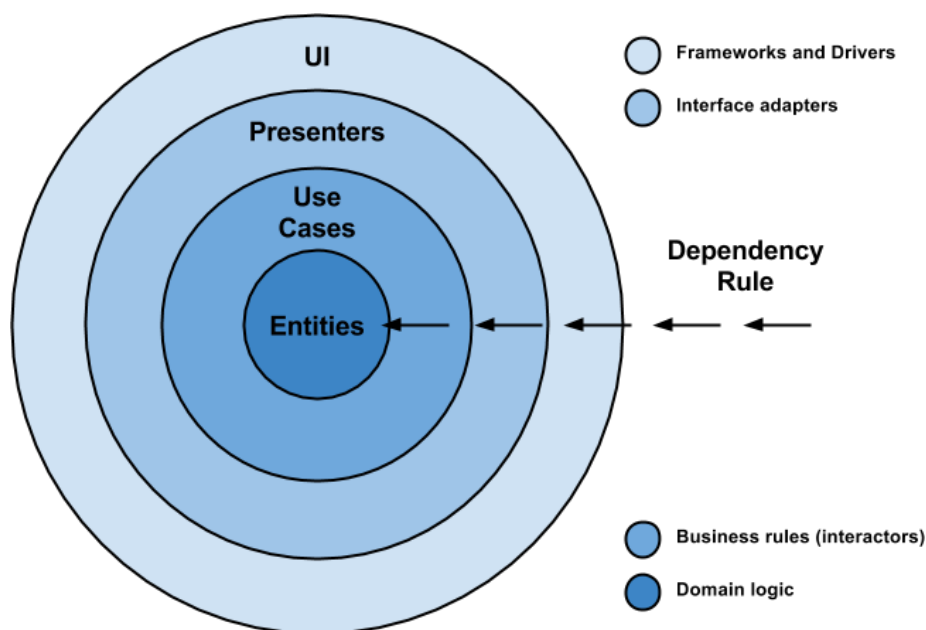


Ilustración 5.2 Dependencias Arquitectura Lógica (Cejas, 2014)

Para mejorar el rendimiento del sistema y evitar bloqueos en la interfaz de usuario, la capa de presentación utilizará el hilo principal del sistema, mientras que las capas de dominio y de data utilizarán un hilo secundario en *background*. De este modo logramos separar los hilos de interfaz de usuario y los de peticiones a bases de datos o servicios que pueden necesitar más tiempo de ejecución. Esto de cara al usuario representará que la pantalla no quede bloqueada mientras se realice una petición.

La capa de presentación:

Donde se realiza la lógica relacionada con las vistas y las vistas utilizando arquitectura MVP (modelo-vista-presentador) dentro de la capa.

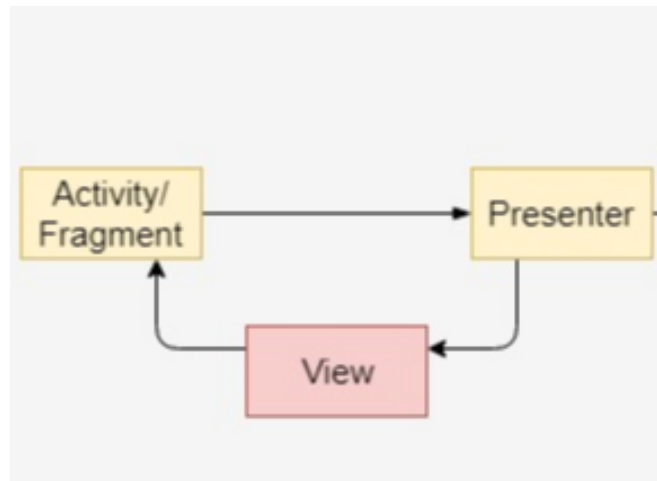


Ilustración 5.3 Capa de presentación

El activity /fragment se tratará de un contenedor visual de Android, cada contenedor visual deberá a su vez tener un presenter asociado encargado de realizar la lógica de la vista, este presenter es inyectado mediante inyección de dependencias en el contenedor visual y no podrá contener elementos de Android. Para permitir la comunicación entre el contenedor presenter y el controlador visual se utilizara una interfaz.

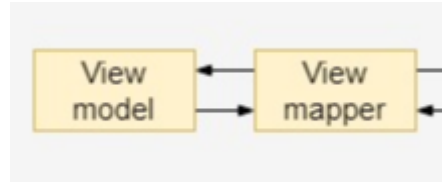


Ilustración 5.4 Modelos y mappers de la capa de presentación

Los modelos de las vistas y los mappers son elementos importantes de la capa, ya que en los modelos de vistas se almacenaran los datos necesarios para las vistas, y el mapper se encargará de transformar objetos de las vista a dominio así como de dominio a vista, permitiendo eliminar elementos innecesarios de una capa a otra.

La capa de dominio:

En esta capa se realiza la lógica de negocio, utilizando únicamente java.

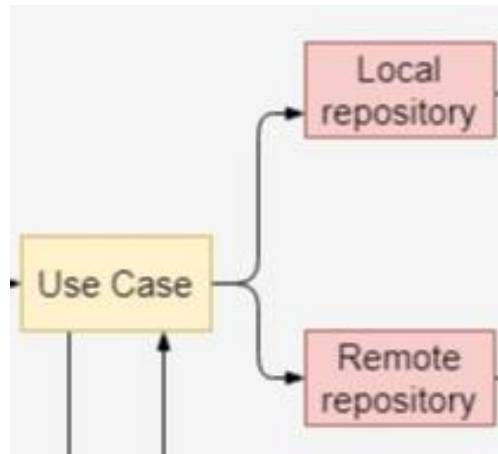


Ilustración 5.5 Capa de dominio

Los casos de uso son llamados por los presenter de la capa de presentación para realizar una acción unitaria que puede requerir de una o varias llamadas a los repositorios, ya sea el repositorio local o el remoto.

Para recibir un elemento de la capa de presentación o de la capa de data estos deberán ser mapeados previamente a un modelo de dominio.

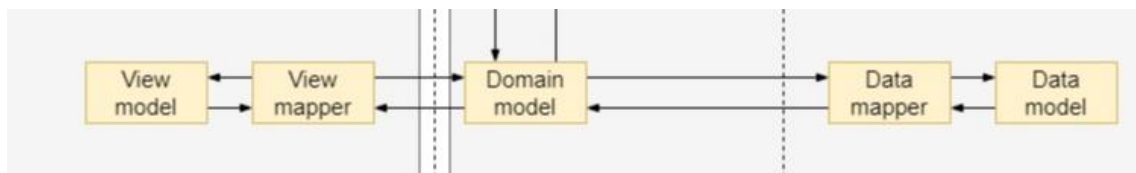


Ilustración 5.6 Modelos y mappers

Mediante los mapper y los distintos modelos según las capas logramos independizar los distintos modelos del resto para no depender de otras capas y responder mejor ante cambios ocurridos en uno de los modelos.

Capa de data:

En esta capa se pueden obtener datos de distintos orígenes sin afectar al resto de capas, en el caso de nuestra aplicación los datos serán obtenidos de una base de datos local mediante Room y de una conexión remota mediante bluetooth utilizando las API de Data API , Message API y Node API.

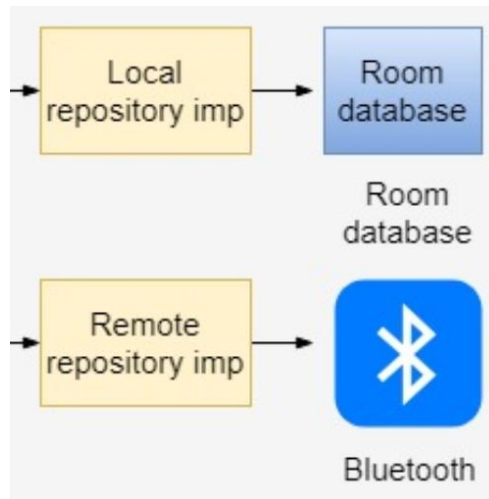


Ilustración 5.7 Capa de data

En el caso de la Room database la arquitectura lógica será la indicada en la siguiente imagen, en la que podemos apreciar el objeto de acceso a datos DAO utilizado para comunicarse con la base de datos SQLite.

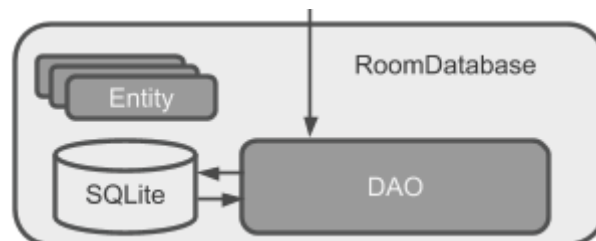


Ilustración 5.8 Room Database

Podemos observar el uso de interfaces al llamar a los repositorios desde la capa de dominio, ya que esto proporciona la posibilidad de crear distintas implementaciones para un tipo de interfaz. Gracias a esto se podría cambiar en cualquier momento la implementación del repositorio sin afectar al resto de la aplicación, permitiendo de esta manera el uso de mock de los datos del repositorio. Para lograr esto, gracias a la inyección de dependencias, bastaría con modificar la implementación inyectada por el provider, para utilizar una implementación del repositorio u otra como podemos observar en la siguiente imagen.

```
@Provides
@Singleton
ConnectRepository provideConnectRepository(ConnectRepositoryImp connectRepositoryImp) {
    return connectRepositoryImp;
}
```

Ilustración 5.9 Provide connect repository

5.2. Arquitectura física

En este apartado se representará la arquitectura física utilizada con sus distintos componentes que forman parte del sistema y sus relaciones.

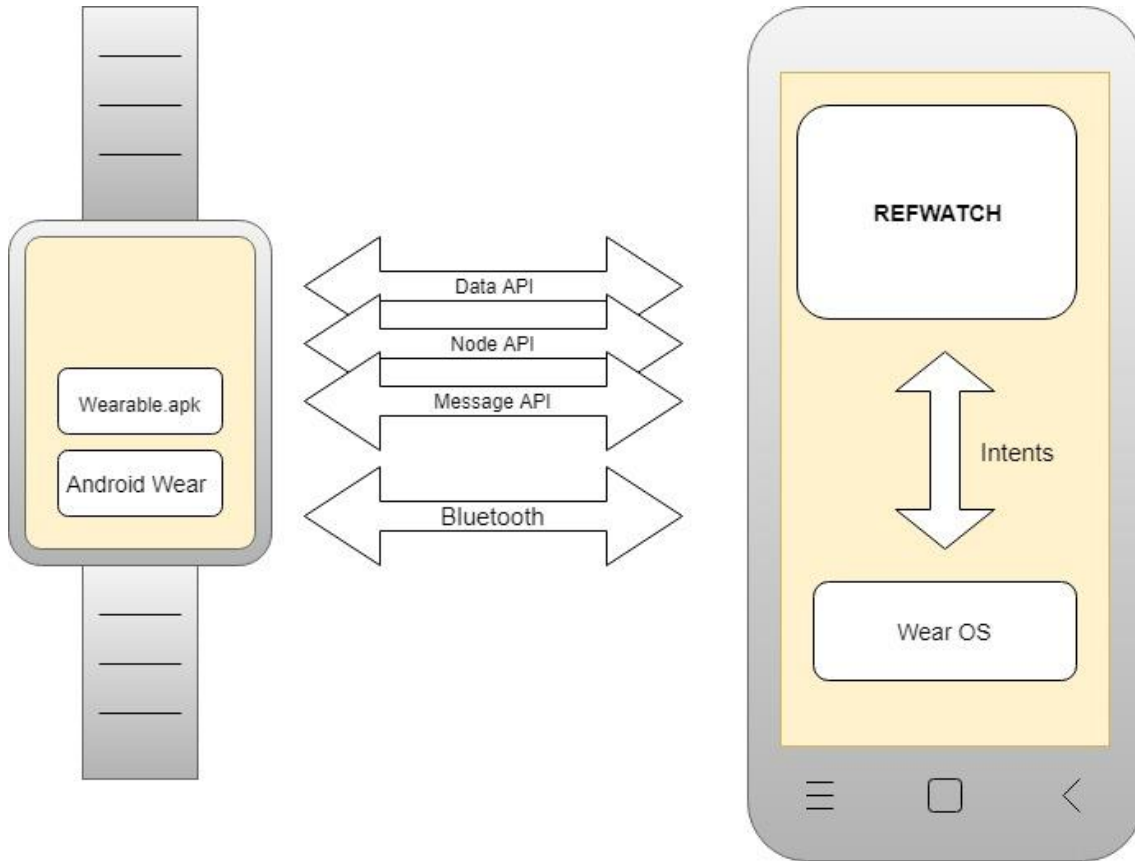


Ilustración 5.10 Arquitectura física

En la imagen de la arquitectura física podemos apreciar la conexión entre el reloj y el teléfono mediante Bluetooth y las distintas API's características de Android Wear.

Data API utilizado para conexiones asíncronas entre el reloj y el teléfono, para no perder mensajes cuando la conexión esta interrumpida.

Message API utilizada para conexiones síncronas entre el reloj y el teléfono.

Node API utilizada para detectar los dispositivos vinculados y mandar datos al dispositivo correcto.

Destaca la necesidad de tener instalada la aplicación de Android Wear ya que hace de intermediario en la conexión entre el teléfono y el reloj en todo momento, por lo cual debemos comunicar datos entre la aplicación de Android Wear y nuestra aplicación.

5.3. Diagrama de secuencia

A partir de los casos de uso crearemos los diagramas de secuencia, los cuales muestran las distintas interacciones entre los componentes del sistema durante la ejecución del mismo.

El siguiente diagrama muestra las interacciones del sistema al guardar un nuevo partido en la base de datos.

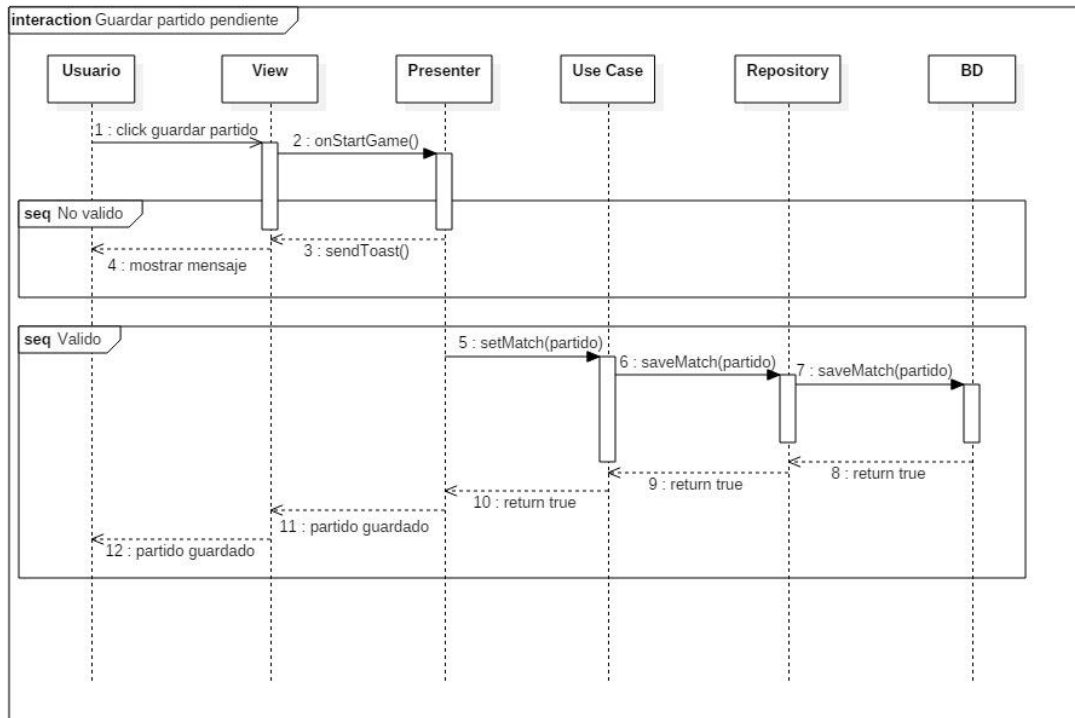


Ilustración 5.11 Diagrama de secuencia Guardar partido pendiente

En el siguiente diagrama se muestran las interacciones del sistema al mostrar el listado de partidos

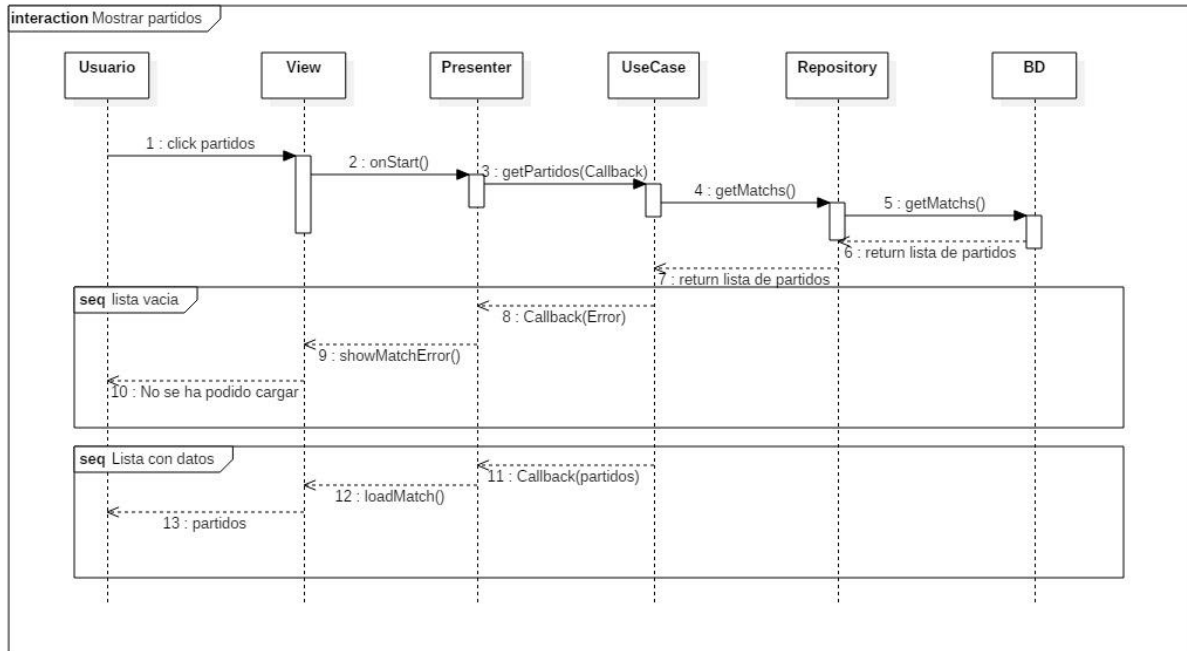


Ilustración 5.12 Diagrama de secuencia Mostrar partidos

En la siguiente imagen se muestran las interacciones del sistema al pulsar sobre el cronómetro

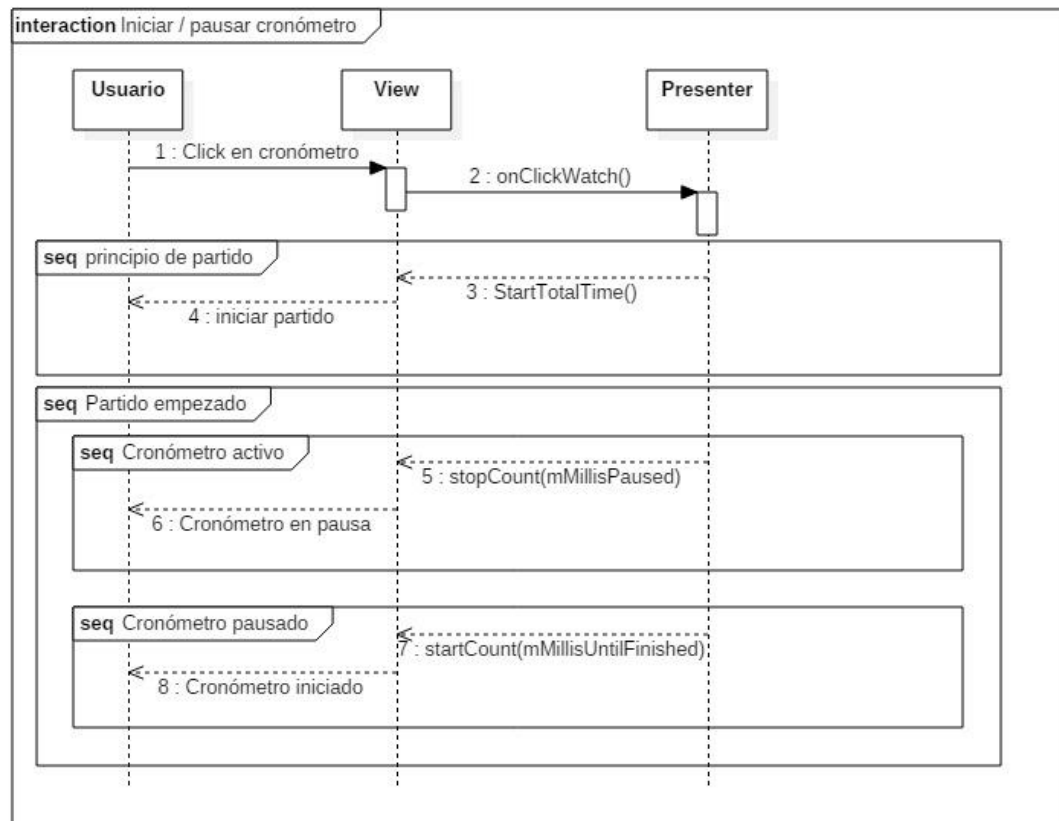


Ilustración 5.13 Diagrama de secuencia Iniciar / pausar partido

5.4. Modelo lógico de datos

El modelo lógico de la aplicación de Android Wear y Android se ha representado con el mismo modelo relacional.

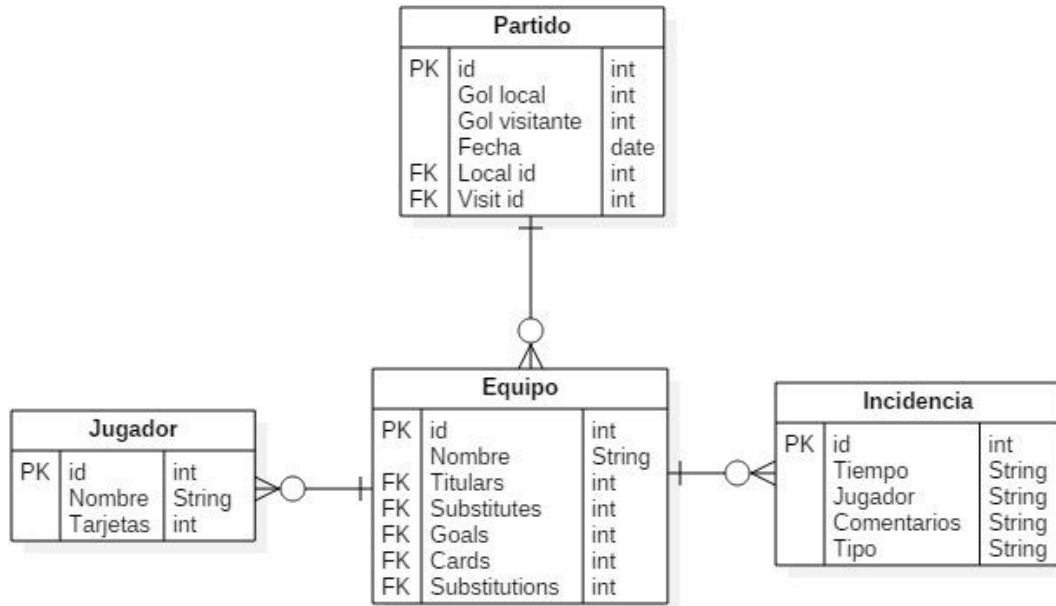


Ilustración 5.14 Modelo Relacional

5.5. Interfaz de usuario

El diseño de la interfaz de usuario se tuvo bastante en cuenta para facilitar su uso.

Al constar de una aplicación wear, el diseño para esta tuvo que simplificar mucho las vistas para facilitar su usabilidad por el tamaño de las pantallas.

Debido a que las pantallas AMOLED son muy frecuentes en este tipo de dispositivo para optimizar el uso de batería, se ha querido hacer énfasis en el ahorro de batería mediante pantallas oscuras que mejoran el uso de batería.

Durante la interfaz de usuario para aplicaciones wear se tuvo que resolver uno de los problemas que tienen las pantallas de los relojes, la forma de la pantalla puede variar bastante de un reloj a otro.

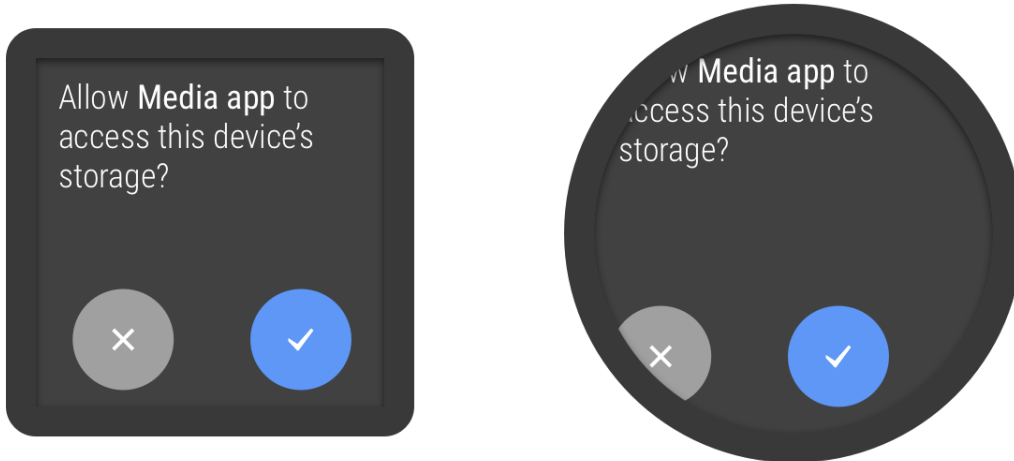


Ilustración 5.15 Problemas de layout en Android Wear (Google, 2018)

Para solucionar esto se utilizó un layout específico de Android wear que permite modificar la pantalla utilizada, para que en el caso de que el reloj disponga de una pantalla circular el layout no tenga problemas.



Ilustración 5.16 Arreglar problema de layout de Android Wear (Google, 2018)

A continuación se explican detalladamente algunas de las pantallas de las aplicaciones móvil y wear:

Aplicación Móvil


Portada	
Descripción	Pantalla de inicio de la aplicación móvil, dispone de 3 botones para permitir al usuario configurar partidos, ver partidos pendientes y ver partidos terminados.
Activación	Inicio de la aplicación
Diseño	 <p style="text-align: center;"><i>Ilustración 5.17 Portada aplicación móvil</i></p>
Eventos	Acceso a las pantallas de configurar partido, partidos pendientes y partidos jugados.

Tabla 37 Portada Aplicación móvil

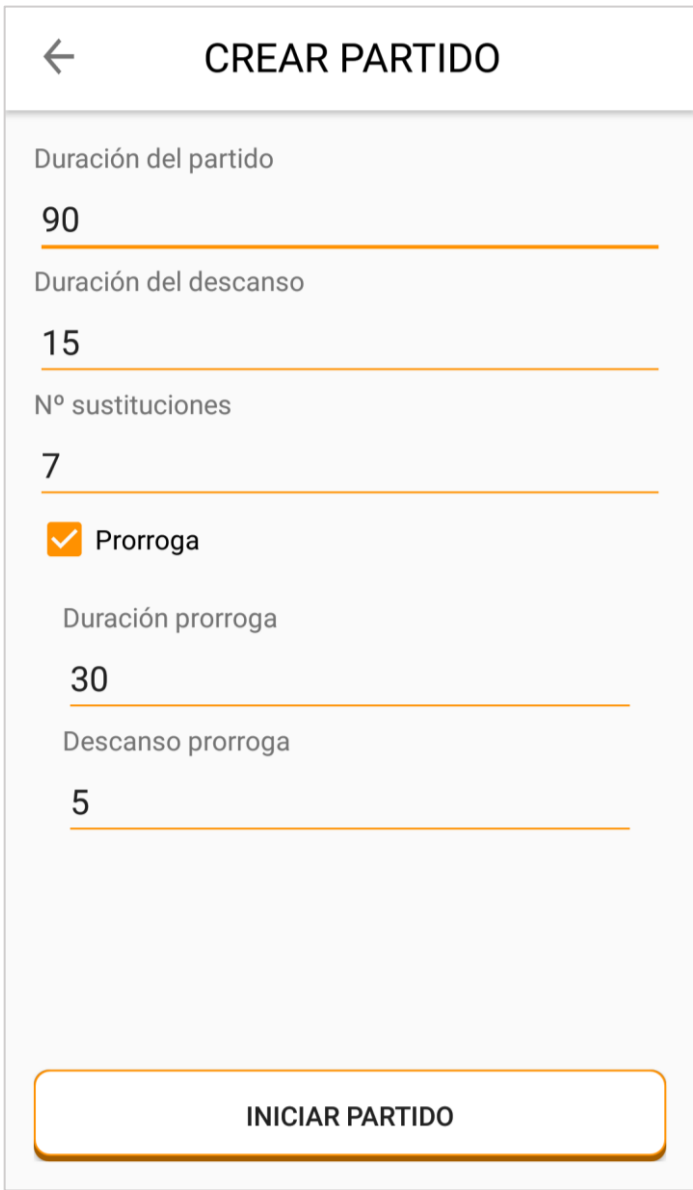
Configurar Partido	
Descripción	Pantalla de configuración de partidos, si prorroga está marcado, se podrán
Activación	Acceso desde la portada de la aplicación
Diseño	 <p style="text-align: center;"><i>Ilustración 5.18 Pantalla configurar partido</i></p>
Eventos	Selección de la configuración del partido

Tabla 38 Pantalla configurar partido

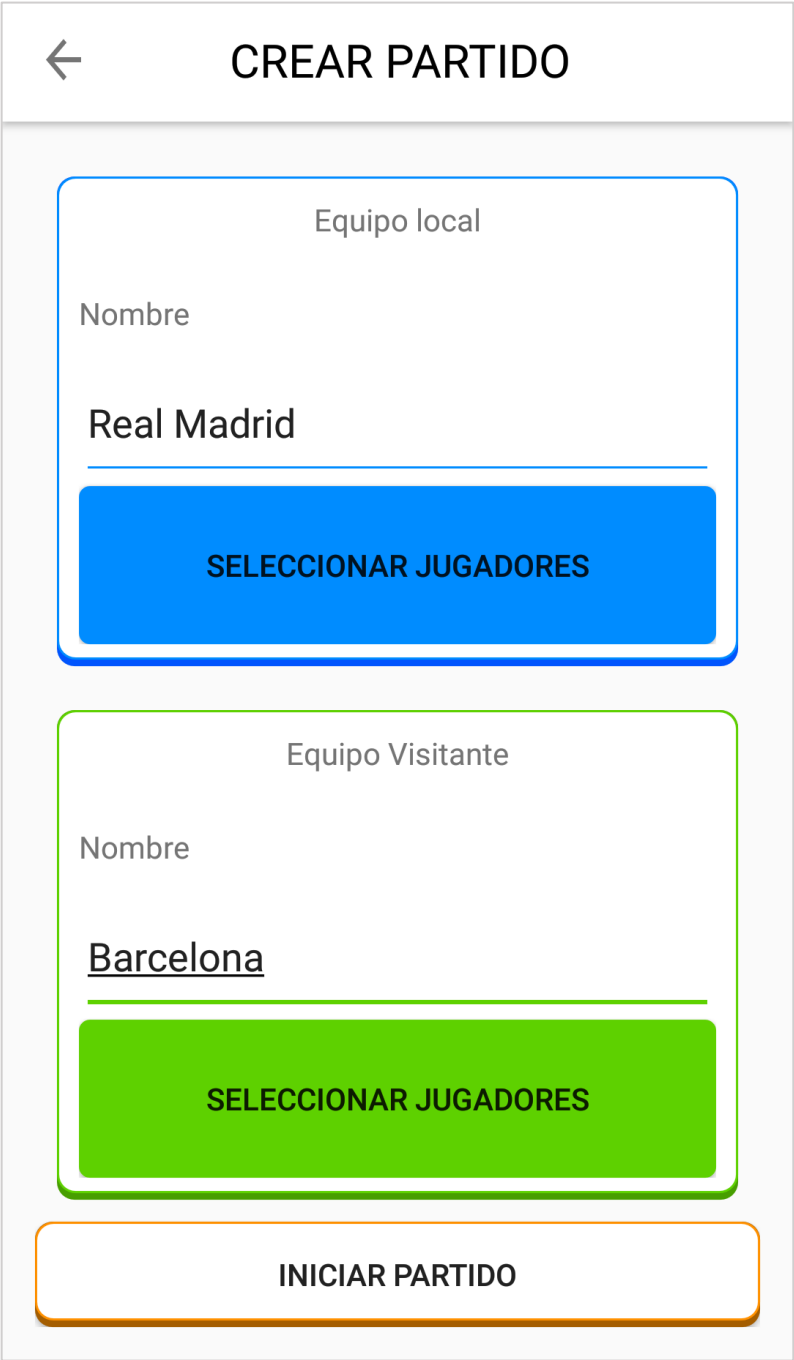
Configurar Equipos	
Descripción	Pantalla de configuración de equipos
Activación	Acceso desde la pantalla de configuración de partidos
Diseño	 <p>The screenshot shows a mobile application screen titled 'CREAR PARTIDO'. At the top left is a back arrow. Below the title, there are two sections for team configuration. The first section is for the 'Equipo local' (Local Team), with a text input field containing 'Real Madrid' and a blue button labeled 'SELECCIONAR JUGADORES'. The second section is for the 'Equipo Visitante' (Visiting Team), with a text input field containing 'Barcelona' and a green button labeled 'SELECCIONAR JUGADORES'. At the bottom of the screen is an orange button labeled 'INICIAR PARTIDO'.</p>
Eventos	Selección de jugadores de los distintos equipos

Ilustración 5.19 Pantalla configurar equipos

Tabla 39 Pantalla configurar equipos

Configurar titulares	
Descripción	Pantalla de selección de jugadores según el equipo seleccionado.
Activación	Acceso desde la pantalla de configuración de equipos
Diseño	 <p style="text-align: center;"><i>Ilustración 5.20 Pantalla configurar titulares</i></p>
Eventos	Acceso a la pantalla de selección de jugadores sustitutos

Tabla 40 Pantalla configurar titulares

Aplicación Wear


Esperando partido	
Descripción	Pantalla de espera de recepción de datos del teléfono móvil
Activación	Tras iniciar la aplicación del dispositivo Android Wear
Diseño	 <p><i>Ilustración 5.21 Pantalla de espera de partido</i></p>
Eventos	Acceso a la pantalla principal de la aplicación tras recibir datos del dispositivo móvil

Tabla 41 Pantalla de espera de partido

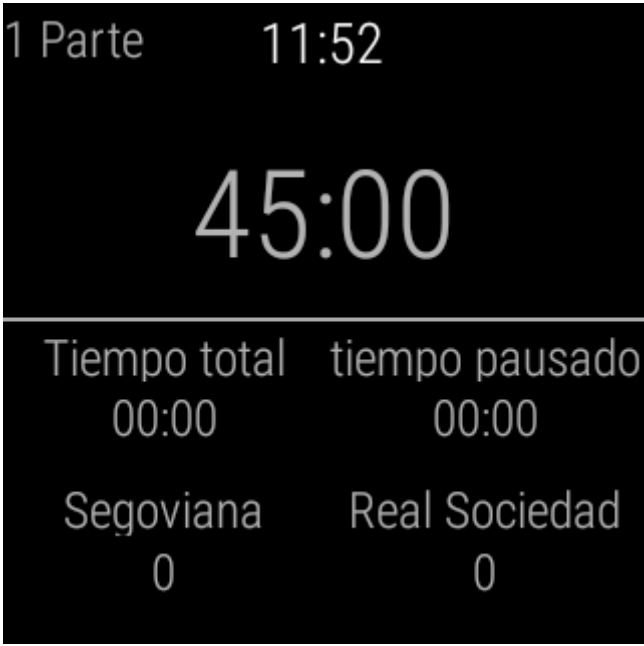
Cronómetro	
Descripción	Pantalla principal de la aplicación Android Wear
Activación	Tras recibir datos del dispositivo móvil con los datos del partido
Diseño	 <p style="text-align: center;"><i>Ilustración 5.22 Pantalla principal de Android Wear</i></p>
Eventos	Acceso a menú desplegable

Tabla 42 Pantalla principal de Android Wear


Añadir gol	
Descripción	Pantalla de añadir goles de la aplicación con dos pestañas según el equipo que ha anotado el gol y un listado de los jugadores de cada equipo.
Activación	Acceso desde el menú desplegable
Diseño	 <p style="text-align: center;"><i>Ilustración 5.23 Pantalla añadir gol</i></p>
Eventos	Añade un gol al equipo elegido y vuelve a la pantalla principal

Tabla 43 Pantalla añadir gol

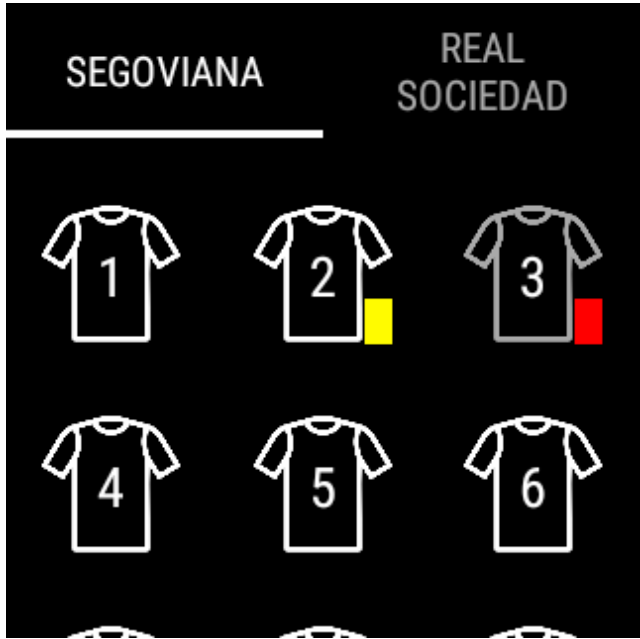
Añadir tarjeta	
Descripción	Permite añadir una tarjeta al jugador elegido y muestra los jugadores con tarjetas previas
Activación	Acceso desde el menú desplegable
Diseño	 <p><i>Ilustración 5.24 Pantalla añadir tarjeta</i></p>
Eventos	Añade un tarjeta al jugador elegido y vuelve a la pantalla principal

Tabla 44 Pantalla añadir tarjeta

Capítulo 6 Implementación

A lo largo de este capítulo se explicarán las diferentes herramientas utilizadas para el desarrollo y análisis de los sistemas Android wear y Android.

6.1. Herramientas de análisis

Word

Programa de edición de texto utilizado para redactar este documento. Presenta una amplia variedad de funcionalidades y facilita en gran medida la realización de documentos gracias a ellas.

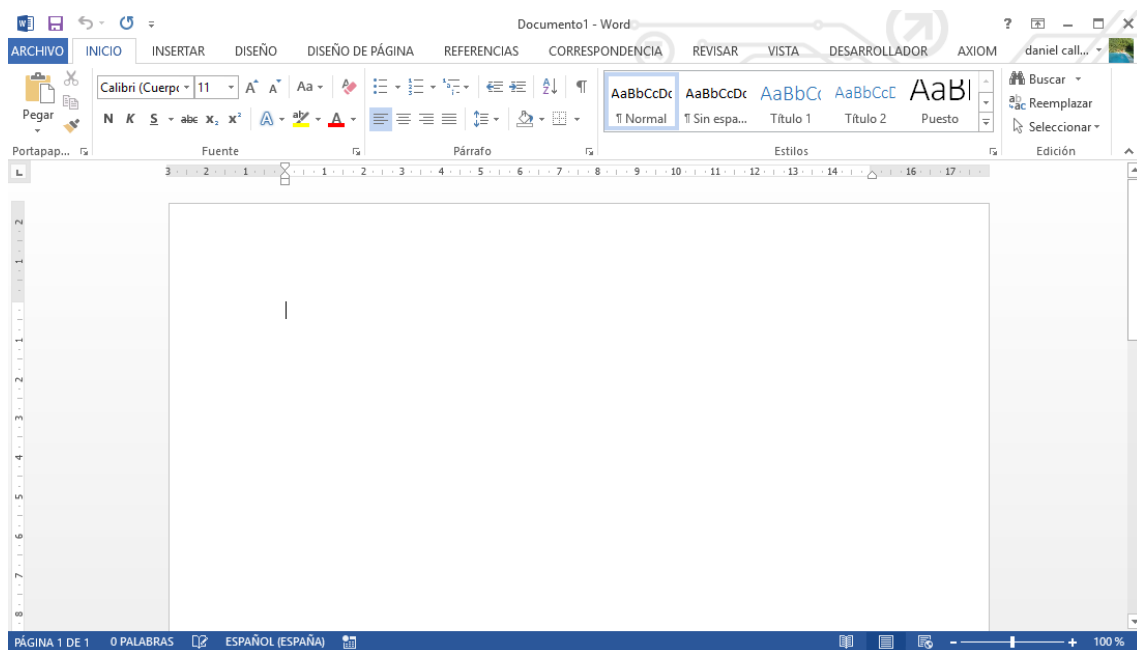


Ilustración 6.1 Word

Draw.io

Herramienta online utilizada para el desarrollo de los diagramas a lo largo del análisis del proyecto. Presenta gran cantidad de figuras para facilitar la realización de diagramas, y su exportación en diversos formatos.

Esta herramienta está ubicada en el dominio <https://www.draw.io/>.

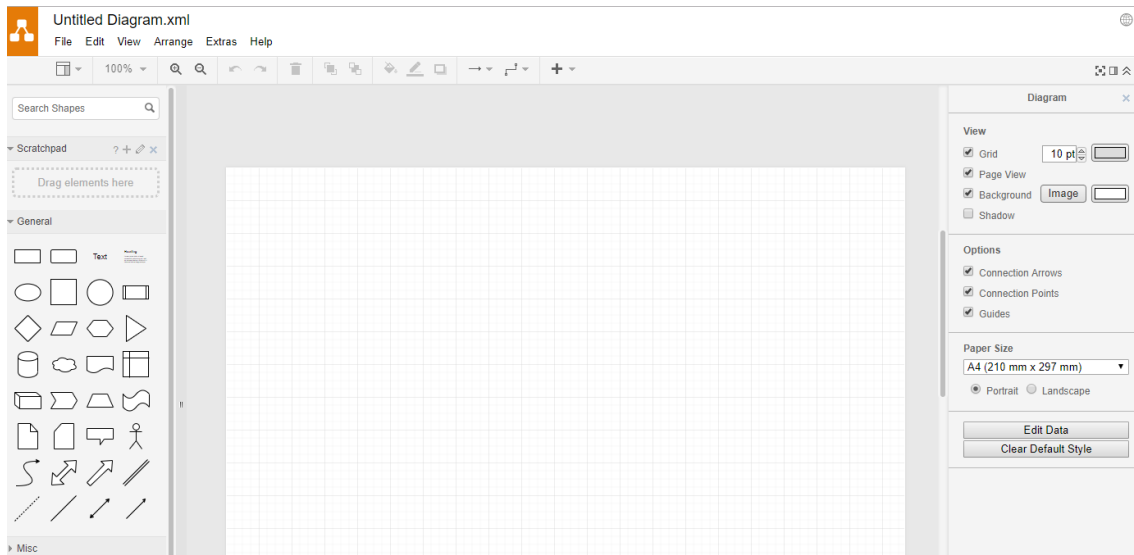


Ilustración 6.2 Draw.io

Android Studio

Entorno de desarrollo oficial de la plataforma Android que tiene gran variedad de plugins para ayudar a la programación, descarga automática de las herramientas necesarias para el desarrollo en función de la versión de Android seleccionada por el desarrollador de la aplicación.

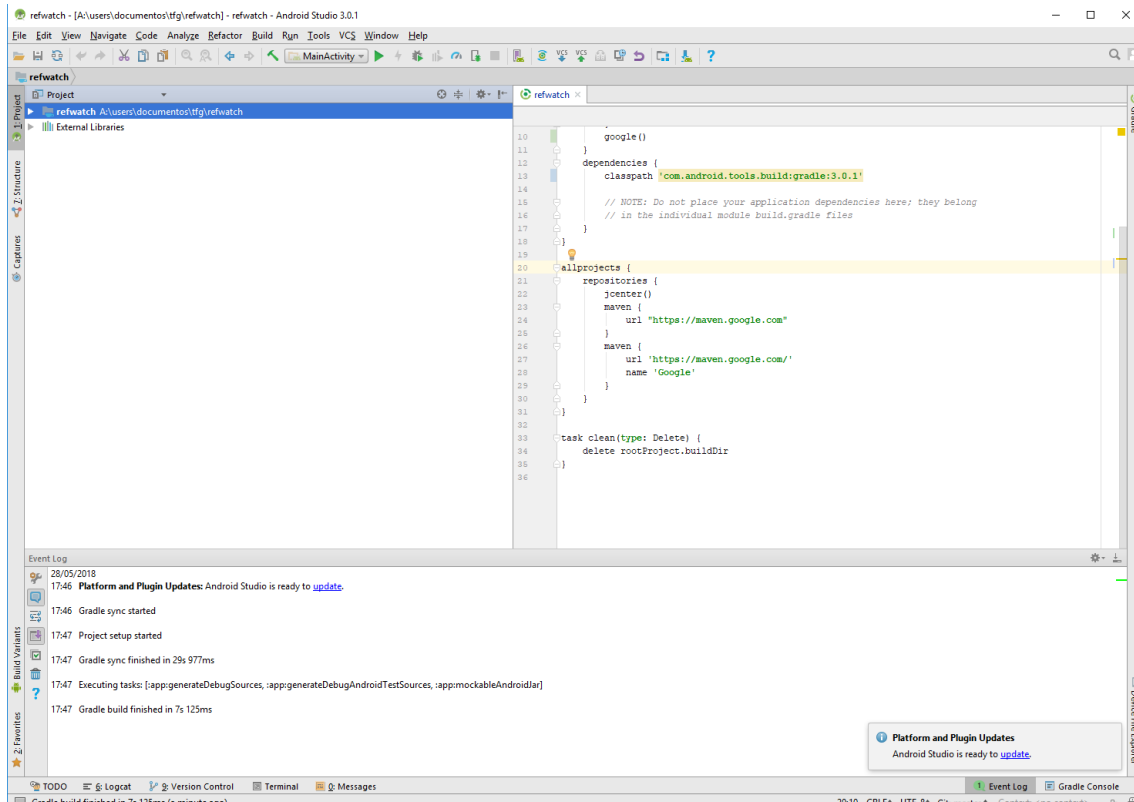


Ilustración 6.3 Android Studio

Android Studio posee de gran cantidad de máquinas virtuales en función de los dispositivos que necesitemos emular durante el desarrollo, en este caso ha sido necesario el uso de una máquina virtual de Android Wear y otra de Android

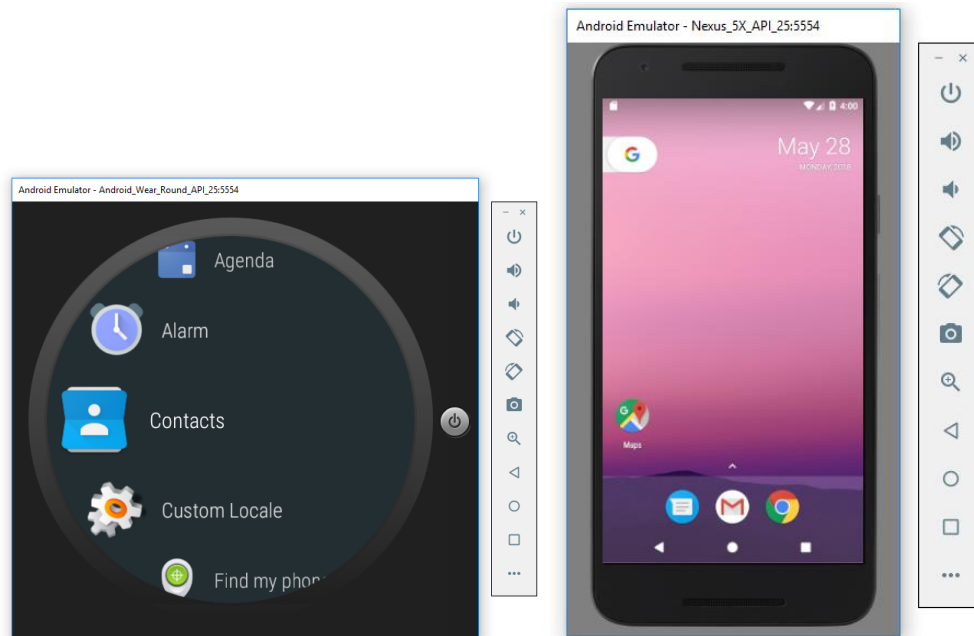


Ilustración 6.4 Máquinas Virtuales de Android Studio

SimpleUMLCE

Plugin para Android Studio que permite generar diagramas UML de manera automática a partir del código fuente de la aplicación.

Para añadir el plugin se debe descargar el fichero jar con la última versión de la página: <https://plugins.jetbrains.com/plugin/4946-simpleumlce>

Para insertar un plugin desde un jar debemos seleccionar install plugin from disk...

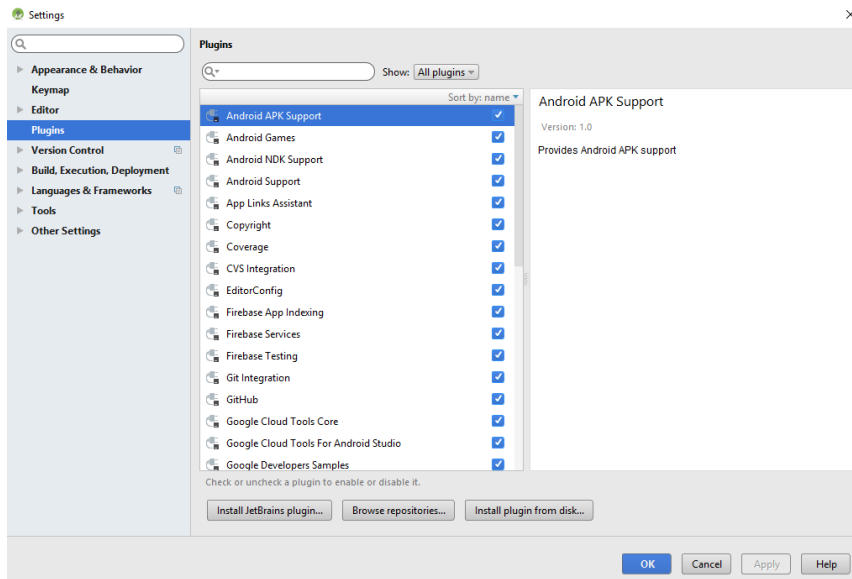


Ilustración 6.5 SimpleUMLCE

Una vez instalado el plugin podremos crear diagramas de manera sencilla seleccionando el proyecto.

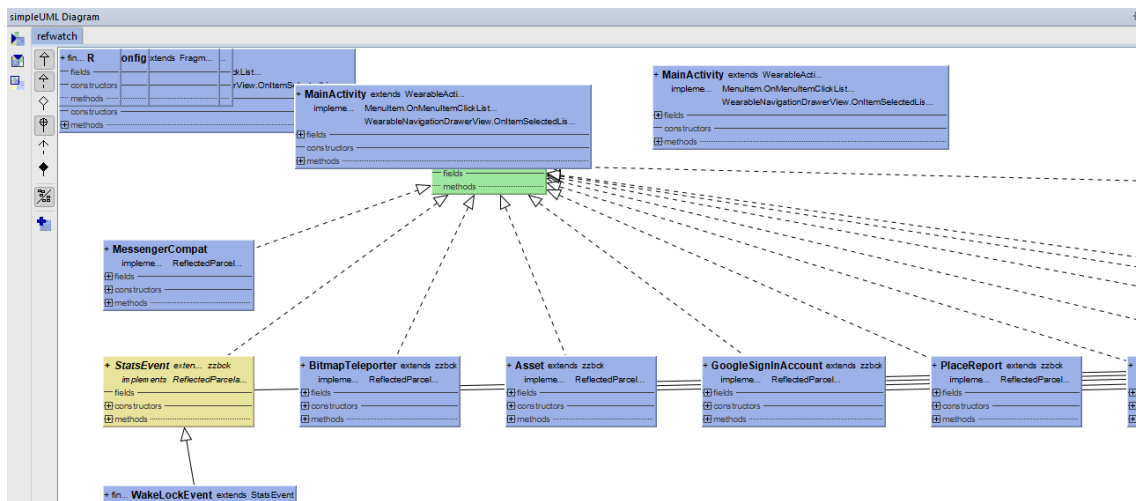


Ilustración 6.6 Uso de SimpleUMLCE

Tomsplanner

Software online utilizado para la planificación del proyecto

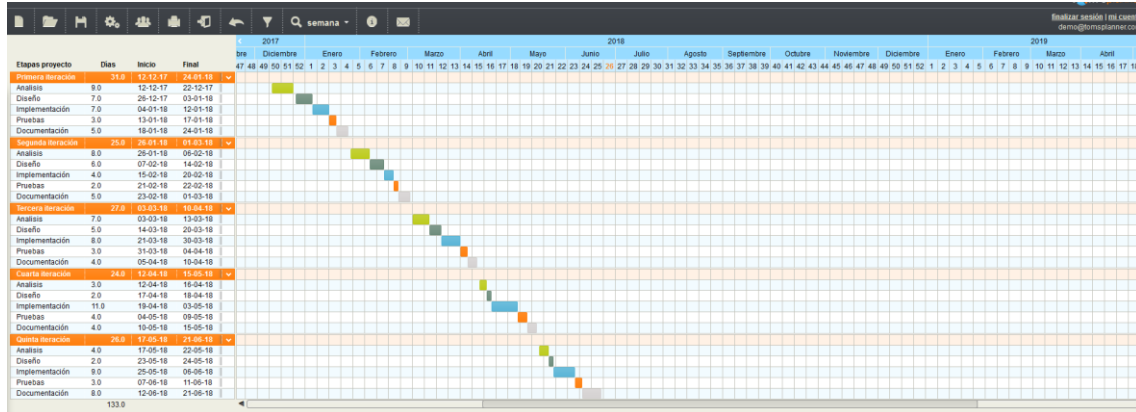


Ilustración 6.7 Tomsplanner

Esta aplicación permite la organización del proyecto en distintos periodos temporales, generando un diagrama de Gantt en función del tiempo especificado en cada una de las iteraciones.

Dispone de las opciones de especificar duración en días de cada etapa, inicio y fin de cada etapa, visualización de distintas maneras, días, semanas o meses. En el caso de este TFG se ha utilizado la visualización por semanas para proporcionar una visión global del tiempo de cada fase del proyecto.

6.2. Herramientas de desarrollo

Room

Librería de persistencia de datos nativa que proporciona una capa de abstracción sobre SQLite para Android.

Esta librería permite crear una cache de los datos de la aplicación en el dispositivo que la ejecuta, lo que permite a la aplicación tener una copia consistente de los datos de la aplicación.

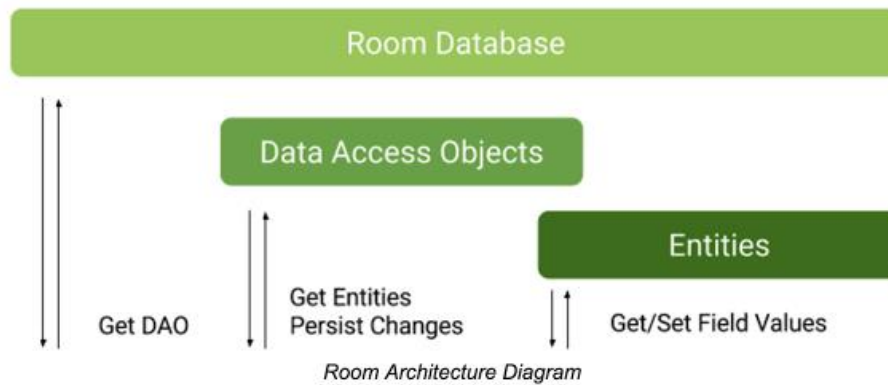


Ilustración 6.8 Room database (Ajay, 2017)

Deberemos añadir las siguientes dependencias en el gradle de nuestro proyecto

```
implementation "android.arch.persistence.room:runtime:1.0.0"
annotationProcessor "android.arch.persistence.room:compiler:1.0.0"
testImplementation "android.arch.persistence.room:testing:1.0.0"
```

Una vez las dependencias estén agregadas necesitaremos crear la base de datos de room para ello podemos hacerlo con el siguiente código:

```
@Database(entities = {DbMatch.class, DbTeam.class}, version = 2)
public abstract class DatabaseReferee extends RoomDatabase {
    public abstract MatchDao matchDao();

    public abstract TeamDao teamDao();

    @Override
    protected SupportSQLiteOpenHelper
    createOpenHelper(DatabaseConfiguration config) {
        return null;
    }

    @Override
    protected InvalidationTracker createInvalidationTracker() {
        return null;
    }
}
```

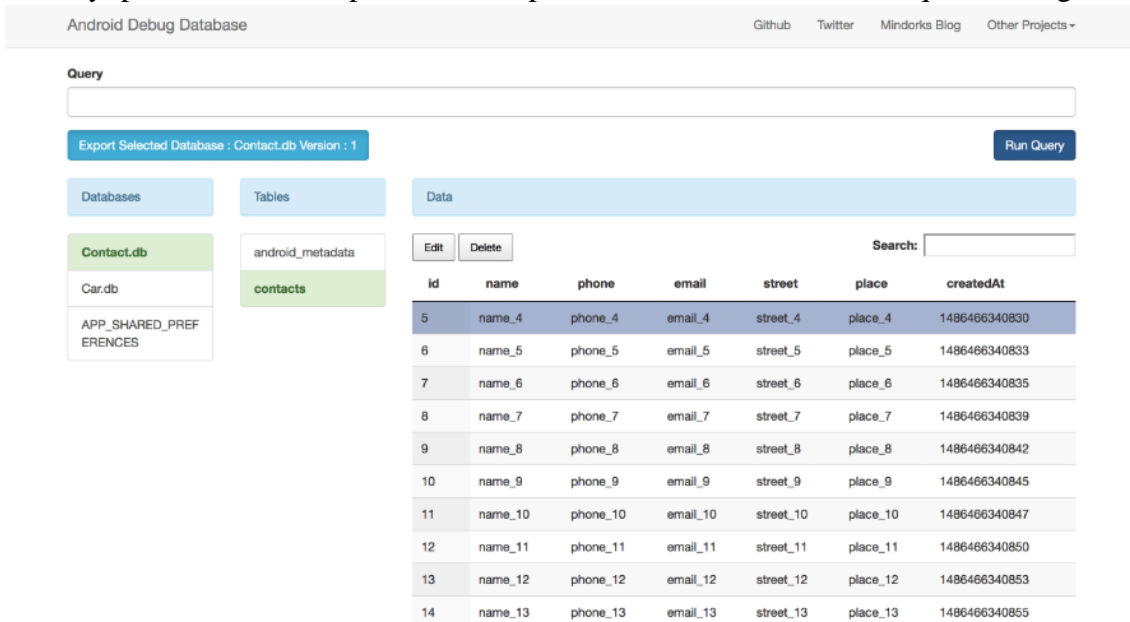
Dicho código llama a una interfaz utilizada para hacer las query como podemos observar en el siguiente fragmento de código:

```
@Dao
public interface MatchDao extends BaseDao<DbMatch> {
    @Query("SELECT * FROM dbmatch")
    List<DbMatch> getAll();

    @Query("SELECT * FROM dbmatch WHERE id=:idMatch")
    public DbMatch loadStatesByServiceId(int idMatch);
}
```


Android Debug Database

Android Debug Database (Shekhar, 2016) es una librería que permite debuggear bases de datos y preferencias compartidas de aplicaciones Android en cualquier navegador.



The screenshot shows the Android Debug Database web interface. At the top, there is a navigation bar with links for GitHub, Twitter, Mindorks Blog, and Other Projects. Below this is a 'Query' input field. A button labeled 'Export Selected Database : Contact.db Version : 1' is visible, along with a 'Run Query' button. The interface is divided into three main sections: 'Databases', 'Tables', and 'Data'. Under 'Databases', 'Contact.db' is selected. Under 'Tables', 'contacts' is selected. The 'Data' section displays a table with columns: id, name, phone, email, street, place, and createdAt. The table contains 10 rows of data, with the first row highlighted in blue. A search bar is located above the table.

id	name	phone	email	street	place	createdAt
5	name_4	phone_4	email_4	street_4	place_4	1486466340830
6	name_5	phone_5	email_5	street_5	place_5	1486466340833
7	name_6	phone_6	email_6	street_6	place_6	1486466340835
8	name_7	phone_7	email_7	street_7	place_7	1486466340839
9	name_8	phone_8	email_8	street_8	place_8	1486466340842
10	name_9	phone_9	email_9	street_9	place_9	1486466340845
11	name_10	phone_10	email_10	street_10	place_10	1486466340847
12	name_11	phone_11	email_11	street_11	place_11	1486466340850
13	name_12	phone_12	email_12	street_12	place_12	1486466340853
14	name_13	phone_13	email_13	street_13	place_13	1486466340855

Ilustración 6.9 Android Debug Database

Para utilizar esta librería deberemos añadir el siguiente código en el fichero build.gradle de nuestra aplicación.

```
debugImplementation 'com.amitshekhar.android:debug-db:1.0.3'
```

Para cambiar el puerto de conexión por un puerto personalizado, tendremos que añadir el siguiente código en el fichero build.gradle

```
debug {  
    resValue("string", "PORT_NUMBER", "8081")  
}
```

Wear OS

Es una aplicación imprescindible para realizar la sincronización de dispositivos y para la utilización del api del Wear.

Permite la conexión con emuladores facilitando la tarea de los desarrolladores a la hora de programar para wear.

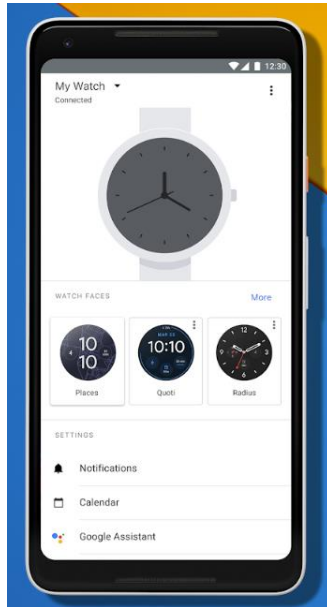


Ilustración 6.10 Wear OS (Google, s.f.)

Wearable API

Para realizar la conexión entre Android wear y un teléfono Android, se requiere el uso de la api de los wearables. Esta API nos permite detectar dispositivos conectados, mandar archivos a otros dispositivos de forma síncrona o asíncrona, según sea requerido por la aplicación.

Para realizar la conexión con el API necesitaremos utilizar el siguiente código:

```
mGoogleApiClient = new GoogleApiClient.Builder(context)
    .addApi(Wearable.API)
    .addConnectionCallbacks(this)
    .addOnConnectionFailedListener(this)
    .build();
mGoogleApiClient.connect();
```

Con este fragmento de código realizaremos la conexión con la API de google.

Para ver el listado de dispositivos conectados utilizaremos la NodeApi:

```
private Collection<String> getNodes() {
    HashSet<String> results = new HashSet<>();
    NodeApi.GetConnectedNodesResult nodes =
    Wearable.NodeApi.getConnectedNodes(mGoogleApiClient).await();
```

```

    for (Node node : nodes.getNodes()) {
        results.add(node.getId());
    }

    return results;
}

```

Gracias a esta API podremos saber los dispositivos a los que estamos conectados en ese momento que utilicen este API.

Para enviar un mensaje a otro dispositivo de manera síncrona utilizaremos Message API como podemos ver en el siguiente fragmento de código:

```

Wearable.MessageApi.sendMessage(
    mGoogleApiClient, node, START_ACTIVITY_PATH,
    bytes).setResultCallback(
    new ResultCallback<SendMessageResult>() {
        @Override
        public void onResult(SendMessageResult sendMessageResult)
        {
            if (!sendMessageResult.getStatus().isSuccess()) {
                Log.e(TAG, "Failed to send message with status
                    code: " +
                sendMessageResult.getStatus().getStatusCode());
                mCanReceive=true;
            }
        }
    }
);

```

Para enviar un mensaje a otro dispositivo de forma asíncrona utilizaremos Data API:

```

Wearable.DataApi.putDataItem(mGoogleApiClient, request)
    .setResultCallback(new ResultCallback<DataItemResult>() {
        @Override
        public void onResult(DataItemResult dataItemResult) {
            Log.d(TAG, "Sending image was successful: " +
                dataItemResult.getStatus()
                    .isSuccess());
        }
    });

```

Dagger2

Es una librería de inyección de dependencias la cual permite:

- Gran flexibilidad de configuración de los objetos.
- Externaliza la configuración de los objetos del propio objeto, permitiendo realizar distintas implementaciones de componentes.
- Facilita el testeo de la aplicación ya que los objetos son más independientes entre sí.
- Fácil de mantener y realizar cambios.
- Gran reutilización de código.
- Disminuye el acoplamiento de una clase y sus dependencias

Para iniciar dagger debemos utilizar el siguiente código en el application de nuestro proyecto:

```
private void initInjector() {
    applicationComponent = DaggerAppComponent.builder()
        .appModule(new AppModule(this))
        .build();
}
```

En cada componente visual donde necesitemos inyectar elementos deberemos inicializar los componentes:

```
public void setupComponent() {
    mFragmentComponent = ((GoalActivity) getActivity()).getComponent()
        .plus(new FragmentModule(this));

    mFragmentComponent.inject(this);
}
```

Cada componente variará en función de su ciclo de vida, el siguiente fragmento sería el código utilizado para un fragment:

```
@FragmentScope
@Subcomponent(modules = FragmentModule.class)
public interface FragmentComponent {

    void inject(MainFragment mainFragment);

    void inject(TeamFragment teamFragment);

    void inject(CardTeamFragment cardTeamFragment);

    void inject(SubTeamFragment subTeamFragment);
}
```

```
@Module
public class FragmentModule {

    private final Fragment mFragment;

    public FragmentModule(Fragment fragment) {
        mFragment = fragment;
    }

    @Provides
    @ActivityScope
    Fragment provideFragment() {
        return this.mFragment;
    }
}
```

Una de las funcionalidades de dagger es la posibilidad de utilizar distintas implementaciones para una misma interfaz.

```

@Provides
@Singleton
ConnectRepository provideConnectRepository(ConnectRepositoryImp
connectRepositoryImp) {
    return connectRepositoryImp;
}

```

En este fragmento de código podemos elegir la implementación que deseemos con tan solo haberla inyectado antes, para lograr esto todos los componentes de la implementación deberán estar inyectados. En el siguiente ejemplo para inyectar correctamente el ConnectRepositoryImp, se debería haber inyectado previamente el contexto en la aplicación.

```

@Inject
public ConnectRepositoryImp(Context context) {
    mContext = context;
}

```

Butterknife

Butterknife (s.f.) es una librería que permite la inyección de vistas y recursos de forma limpia

```

class ExampleActivity extends Activity {
    @BindView(R.id.user) EditText username;
    @BindView(R.id.pass) EditText password;

    @BindString(R.string.login_error) String loginErrorMessage;

    @OnClick(R.id.submit) void submit() {
        // TODO call server...
    }

    @Override public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.simple_activity);
        ButterKnife.bind(this);
        // TODO Use fields...
    }
}

```

Para utilizar butterknife deberemos asegurarnos de añadir las siguientes dependencias en el gradle del proyecto:

```

annotationProcessor 'com.jakewharton:butterknife-compiler:8.8.1'
compile 'com.jakewharton:butterknife:8.8.1'

```

Gson

Librería Java que permite el serializado y deserializado entre Objetos java y Json de manera sencilla.

Para utilizar gson deberemos agregar la siguiente dependencia en el gradle del proyecto:

```
compile 'com.google.code.gson:gson:2.8.2'
```

Para convertir objetos a Json utilizaremos el siguiente código:

```
Gson gson = new Gson();  
String match = gson.toJson(matchDomain);
```

Para convertir de Json a objetos utilizaremos el siguiente código:

```
Gson gson = new Gson();  
ClockModel match = gson.fromJson(text, ClockModel.class);
```

Mockito

Librería utilizada para la realización de test unitarios e instrumentales para Android.

Durante las pruebas de caja negra de la aplicación se utilizará para realizar test instrumentales.

Para utilizarlo primero deberemos añadir las dependencias:

```
testCompile 'junit:junit:4.12'  
testCompile 'org.mockito:mockito-core:2.11.0'  
testCompile 'org.hamcrest:hamcrest-library:1.3'
```

Una vez añadidas las dependencias podremos grabar el funcionamiento de la aplicación, para ello seleccionamos la opción record Espresso Test dentro del menú de Run

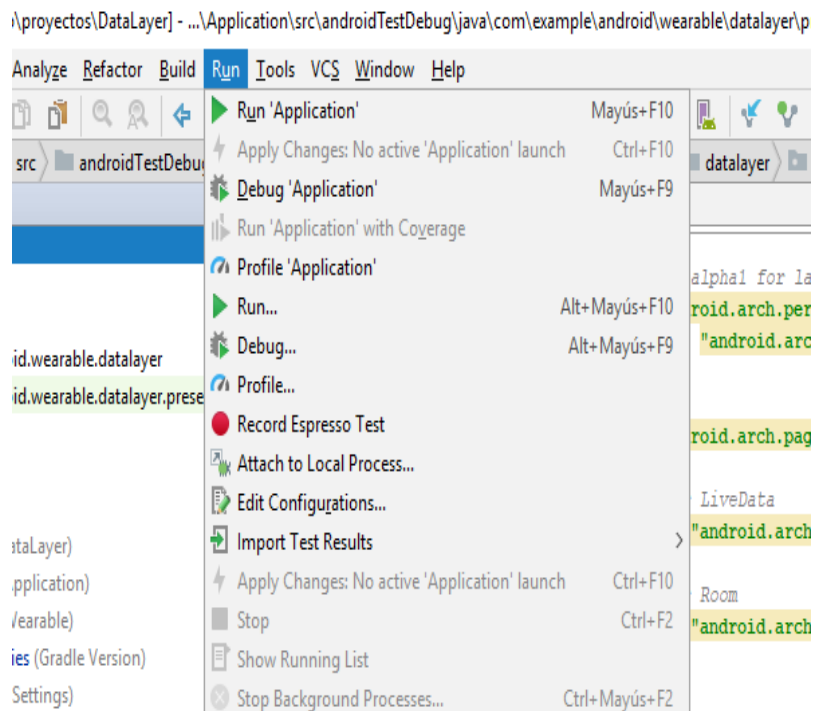


Ilustración 6.11 Start Espresso test

Una vez iniciada la grabación podremos realizar las acciones que queramos grabar durante el test y al finalizar de grabar se creará automáticamente el test deseado con las acciones previas realizadas. Este tipo de test es útil para reproducir resultados rápidamente, ya que está programado para realizar una serie de acciones previamente determinadas

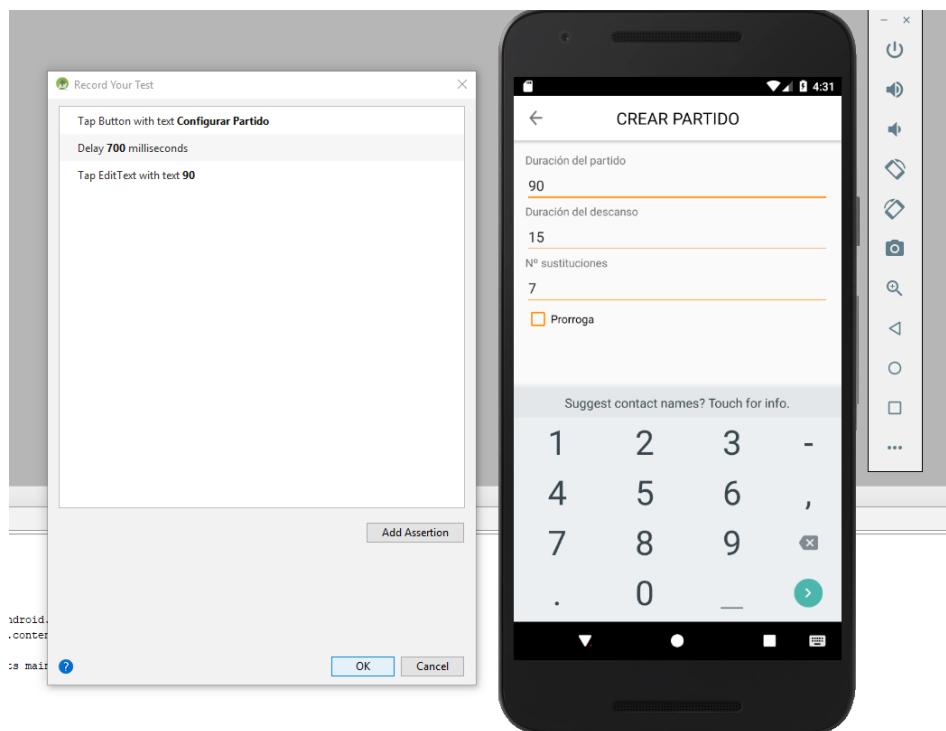


Ilustración 6.12 Grabar test con Espresso

Capítulo 7 Pruebas

En este capítulo se describirán las pruebas realizadas sobre los dos sistemas. Se han realizado dos tipos de pruebas, pruebas de caja blanca y pruebas de caja negra.

7.1. Pruebas de caja blanca

Las pruebas de caja blanca son pruebas de bajo nivel sobre el funcionamiento independiente de cada una de las aplicaciones en distintas situaciones. Para realizar estas pruebas se ha finalizado previamente la funcionalidad para testear el código recorriendo todos los posibles caminos que pueda recorrer durante la ejecución. A continuación se listan las comprobaciones realizadas:

Caja blanca 1: Verificar configuración de jugadores titulares

```
1 @OnClick(R.id.bt_selectplayers_next)
  void onToSelectSubs() {
    mPresenter.toSelectSubs(adapter.getCheckedPlayers());
    mPlayers = (ArrayList) adapter.getPlayerList();
  }
```

```
2 public List<Player> getCheckedPlayers() {
  List<Player> selectedPlayers = new ArrayList<>();
  for (Player player : playerList) {
3     if (player.isChecked()) {
4         selectedPlayers.add(player);
        }
    }
  return selectedPlayers;
}
```

```
5 public void toSelectSubs(List<Player> checkedPlayers) {
  if (checkedPlayers.size() >= 7)
6     mView.toSelectSubs(checkedPlayers);
  else
    mView.needMorePlayers();
}
```

```
@Override
public void needMorePlayers() {
  Toast toast = Toast.makeText(
  getActivity().getApplicationContext(),
  R.string.minPlayers,
  Toast.LENGTH_LONG);
}
```

7

```
toast.show();
}
@Override
public void toSelectSubs(List<Player> checkedPlayers) {
    SelectSubFragment fragment =
    SelectSubFragment.newInstance(checkedPlayers, mPlayers,
    getArguments().getInt(N_SUBS));
    getFragmentManager()
        .beginTransaction()
        .replace(R.id.container_master, fragment)
        .commit();
}
```

Diagrama de flujo del método:

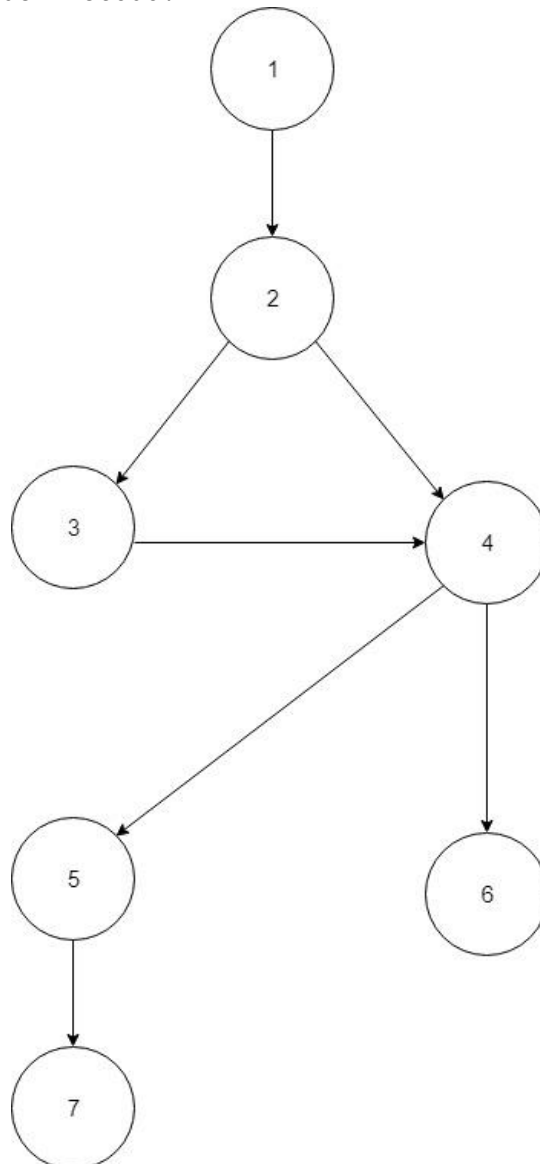


Ilustración 7.1 Diagrama de flujo de caja blanca 1

Diseño y realización de pruebas:

La complejidad ciclomática de este sistema será de 3, por lo que tendrá 3 posibles caminos de pruebas

Camino 1: 1, 2, 3, 4, 5, 7.

Camino 2: 1, 2, 3, 4, 6.

Camino 3: 1, 2, 4, 6

Prueba 1	Camino 1
Jugadores seleccionados	2, 4, 5, 6, 7, 10, 15, 16, 20
Salida obtenida	Continua con la ejecución de la aplicación y pasa a la siguiente pantalla

Tabla 45 Prueba de caja blanca 1

Prueba 2	Camino 2
Jugadores seleccionados	2, 4, 5, 6
Salida obtenida	Muestra un mensaje de error avisando al usuario de que se necesitan al menos 7 jugadores.

Tabla 46 Prueba de caja blanca 2

Prueba 3	Camino 3
Jugadores seleccionados	Null
Salida obtenida	Muestra un mensaje de error avisando al usuario de que se necesitan al menos 7 jugadores.

Tabla 47 Prueba de caja blanca 3

Caja blanca 2: Añadir gol

```
1  @Override
   public void onTabSelected(TabLayout.Tab tab) {
       // get the current selected tab's position and replace the
       // fragment accordingly
       switch (tab.getPosition()) {
2     case 0:
           replaceFragment(fragmentLocal);
           break;
3     case 1:
           replaceFragment(fragmentVisitant);
           break;
       }
   }
```

```
4  private void remplazeFragment(Fragment fragment) {
       FragmentManager fm = getFragmentManager();
       FragmentTransaction ft = fm.beginTransaction();
       ft.replace(R.id.simpleFrameLayout, fragment);
       ft.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_OPEN);
       ft.commit();
   }
```

```
@OnClick(R.id.bt_player)
void onMatchClick() {
    mTeamPresenter.onPlayerSelected(btPlayer.getText().toString());
}
```

```

public void onPlayerSelected(String text) {
    mPresenter.onPlayerSelected(text, mTeam);
}

```

```

5 public void onPlayerSelected(String text, boolean team) {
6     if (mIsAutogoal) {
7         mView.askType(text, team);
8     } else {
9         mView.endActivityForResult(text, team, "Gol");
10    }
11 }

```

```

7 @Override
public void endActivityForResult(String text, boolean team,
String type) {
    Intent data = new Intent();
    data.putExtra(NumberPlayer, text);
    data.putExtra(Team, team);
    data.putExtra(Type, type);
    setResult(RESULT_OK, data);
    finish();
}

```

```

8 @Override
9 public void askType(String text, boolean team) {
10     new AlertDialog.Builder(this)
11         .setTitle("Selecciona tipo")
12         .setPositiveButton("AutoGol", (dialog1, which) ->
13             mPresenter.onTypeAutogoal(text, team))
14         .setNegativeButton("Penal", (dialog1, which) ->
15             mPresenter.onTypePenal(text, team))
16         .show();
17 }

```

```

public void onTypePenal(String text, boolean team) {
    mView.endActivityForResult(text, team, "Penal");
}

```

```

11 public void onTypeAutogoal(String text, boolean team) {
    mView.endActivityForResult(text, team, "Autogol");
}

```

Diagrama de flujo del método:

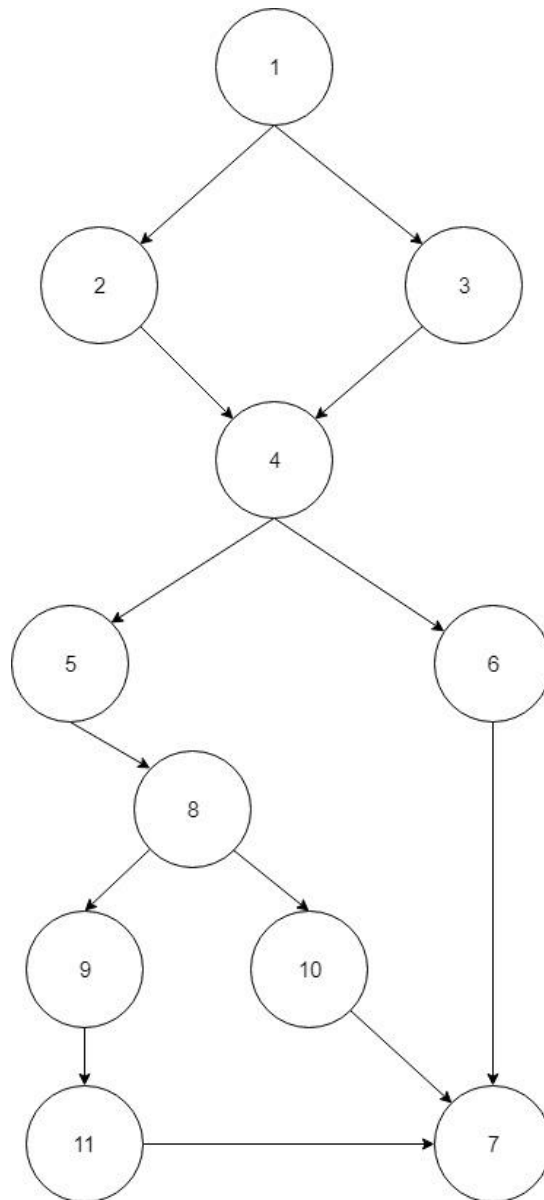


Tabla 48 Diagrama de flujo de caja blanca 2

Diseño y realización de pruebas:

La complejidad ciclomática de este sistema será de 6, por lo que tendrá 6 posibles caminos de pruebas

Camino 1: 1, 2, 4, 6, 7.

Camino 2: 1, 3, 4, 6, 7.

Camino 3: 1, 2, 4, 5, 8, 10, 7.

Camino 4: 1, 3, 4, 5, 8, 10, 7.

Camino 5: 1, 2, 4, 5, 8, 9, 11, 7

Camino 6: 1, 3, 4, 5, 8, 9, 11, 7

Prueba 1	Camino 1
Equipo	Local
Tipo	Gol
Salida obtenida	Se añade un gol de tipo gol en el equipo local

Tabla 49 Prueba de caja blanca 1

Prueba 2	Camino 2
Equipo	Visitante
Tipo	Gol
Salida obtenida	Se añade un gol de tipo gol en el equipo visitante

Tabla 50 Prueba de caja blanca 2

Prueba 3	Camino 3
Equipo	Local
Tipo	Penal
Salida obtenida	Se añade un gol de tipo penal en el equipo local

Tabla 51 Prueba de caja blanca 3

Prueba 4	Camino 4
Equipo	Visitante
Tipo	Penal
Salida obtenida	Se añade un gol de tipo penal en el equipo visitante

Tabla 52 Prueba de caja blanca 4

Prueba 5	Camino 5
Equipo	Local
Tipo	Autogol
Salida obtenida	Se añade un gol de tipo autogol en el equipo local

Tabla 53 Prueba de caja blanca 5

Prueba 6	Camino 6
Equipo	Visitante
Tipo	Autogol
Salida obtenida	Se añade un gol de tipo autogol en el equipo visitante

Tabla 54 Prueba de caja blanca 6

7.2. Pruebas de caja negra

Las pruebas de caja negra o pruebas funcionales son utilizadas para comprobar que las funcionales del sistema se adecuan a los requisitos establecidos previamente. Para su realización se han utilizado test instrumentales mediante Android Studio con el fin de reproducir las distintas pruebas de caja negra utilizadas.

A continuación se mostrarán las pruebas más representativas de cada uno de los sistemas:

PN-01: Configurar Partido	
Objetivo	Verificar que se puede añadir la configuración de un partido
Precondiciones	Ninguna
Datos de entrada	Duración del partido: 90 Duración del descanso : 15 Número de sustituciones: 7 Prorroga: true Duración prorroga: 30 Duración del descanso de la prorroga: 5
Acción esperada	Se muestra la configuración de los equipos
Secuencia	<ol style="list-style-type: none"> 1. Completar duración del partido. 2. Completar duración del descanso. 3. Completar número de sustituciones. 4. Completar prorroga 5. Completar duración prorroga 6. Completar duración prorroga 7. Hacer clic sobre “Configurar Equipos”
Resultado	Correcto

Tabla 55 PN-01: Configurar Partido

PN-02: Configurar Equipos	
Objetivo	Verificar la configuración de los equipos
Precondiciones	Se deben haber fijado previamente los datos de configuración del partido
Datos de entrada	Nombre equipo local: Real Madrid Jugadores titulares equipo local: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 Jugadores suplentes equipo local: 16, 17, 18, 19, 22 Nombre equipo Visitante: FB Barcelona Jugadores titulares equipo Visitante: 1, 2, 3, 8, 9, 10, 11, 12, 13, 14, 22 Jugadores suplentes equipo Visitante: 4, 5, 6, 7, 16
Acción esperada	El partido quedará guardado en la base de datos de la aplicación
Secuencia	<ol style="list-style-type: none"> 1. Completar el nombre del equipo local. 2. Seleccionar los jugadores titulares del equipo local. 3. Seleccionar los jugadores suplentes del equipo local. 4. Completar el nombre del equipo visitante. 5. Seleccionar los jugadores titulares del equipo visitante. 6. Seleccionar los jugadores suplentes del equipo visitante. 7. Hacer clic sobre “Guardar Partido”
Resultado	Correcto

Tabla 56 PN-02: Configurar Equipos

PN-03: Ver partidos pendientes	
Objetivo	Comprobar que se muestran correctamente los partidos pendientes
Precondiciones	Debe haber al menos un partido guardado en la base de datos
Datos de entrada	Ninguno
Acción esperada	Se listarán los partidos almacenados en la base de datos
Secuencia	Tras seleccionar en la portada de la aplicación “Partidos Pendientes”, se mostrará un listado de los partidos pendientes.
Resultado	Correcto.

Tabla 57 PN-03: Ver partidos pendientes

PN- 04: Recibir partido	
Objetivo	Recibir los datos del partido y mostrar los datos.
Precondiciones	Recibir un partido
Datos de entrada	<p>Id del partido: 1</p> <p>Duración del partido: 90</p> <p>Duración del descanso: 15</p> <p>Prorroga: true</p> <p>Duración prorroga: 30</p> <p>Duración del descanso de la prorroga: 5</p> <p>Nombre equipo local: Real Madrid</p> <p>Jugadores titulares equipo local: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11</p> <p>Jugadores suplentes equipo local: 16, 17, 18, 19, 22</p> <p>Nombre equipo Visitante: FB Barcelona</p> <p>Jugadores titulares equipo Visitante: 1, 2, 3, 8, 9, 10, 11, 12, 13, 14, 22</p> <p>Jugadores suplentes equipo Visitante: 4, 5, 6, 7, 16</p>
Acción esperada	Iniciar datos del partido.
Secuencia	<ol style="list-style-type: none"> 1. Se reciben los datos del partido. 2. Configuramos los cronómetros en función de los tiempos recibidos del partido. 3. Configuramos los nombres de los equipos en función de los datos recibidos. 4. Permitimos que el partido se pueda iniciar.
Resultado	Correcto

Tabla 58 PN- 04: Recibir partido

PN-05: Iniciar/pausar cronómetro	
Objetivo	Iniciar o pausar el cronometro
Precondiciones	El partido debe de estar iniciado
Datos de entrada	Ninguno
Acción esperada	Si el cronómetro principal esta pausado: <ul style="list-style-type: none"> • El cronómetro principal continuará. • El cronómetro de pausa se detendrá. Si el cronómetro principal está en funcionamiento: <ul style="list-style-type: none"> • El cronómetro principal se pausará. • El cronómetro de pausa continuará.
Secuencia	1. Pulsar sobre el cronómetro principal
Resultado	Correcto

Tabla 59 PN-05: Iniciar/pausar cronómetro

PN-06: Añadir gol	
Objetivo	Añadir un gol para el equipo local
Precondiciones	Debe estar iniciado el partido. No se debe de haber terminado el periodo.
Datos de entrada	Equipo seleccionado: local Jugador seleccionado: 4
Acción esperada	Se añadirá un gol para el equipo local y se actualizará el marcador en la pantalla principal del reloj.
Secuencia	1. El usuario seleccionar “Gol” en el menú principal de la aplicación. 2. El usuario selecciona el equipo que ha anotado el gol. 3. El usuario selecciona el jugador que ha anotado el gol.
Resultado	Correcto

Tabla 60 PN-06: Añadir gol

Capítulo 8 Manuales

En este capítulo se explicará cómo realizar la instalación de los sistemas, tanto el Android Wear como el Android, y de cómo utilizar la aplicación.

8.1. Manual de instalación

Ambos sistemas tienen un APK de instalación del mismo, para realizar la instalación la realizaremos de distinta manera, pero en ambos casos se requerirá de tener Android studio instalado para realizar la instalación.

Para realizar la instalación en nuestro Android Wear deberemos seguir una serie de pasos desde nuestro reloj:

1. En el menú de configuración de nuestro dispositivo seleccionaremos la opción Acerca de.
2. Deberemos pulsar sobre el Build Number repetidas veces hasta que se nos notifique que se han activado los permisos de desarrollo.

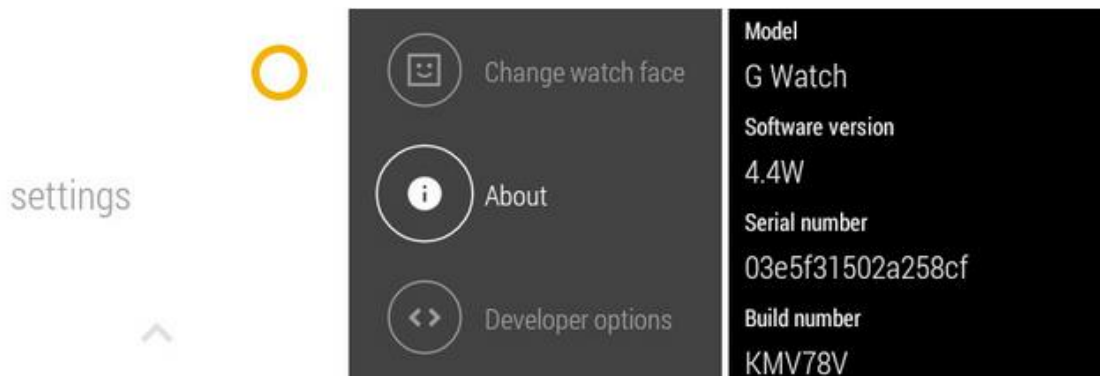


Ilustración 8.1 Activar opciones de desarrollo en Android Wear (Subirats, 2014)

3. Una vez están habilitadas las opciones de desarrollo podremos conectar el teléfono al ordenador, para ello tendremos que ir a las opciones de desarrollo y habilitar la depuración ADB y la depuración por WI-FI, fijándonos en la IP y puerto indicados para utilizarlos más adelante.

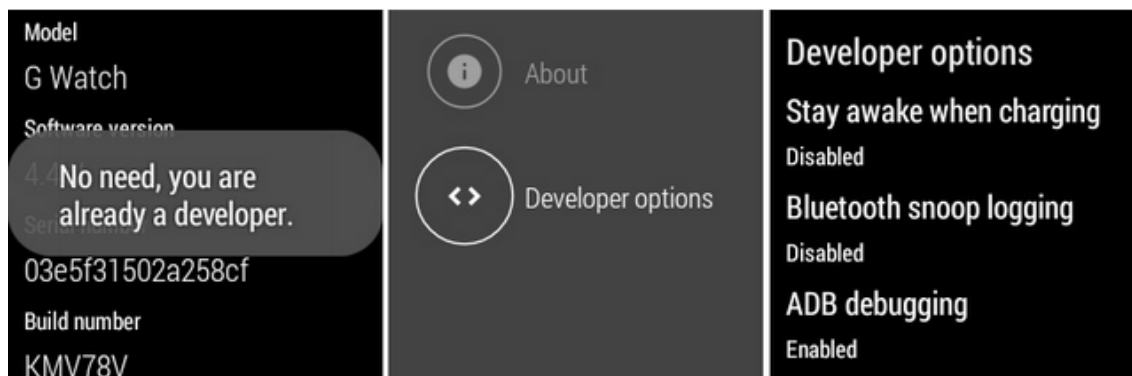


Ilustración 8.2 Activar depuración WIFI Android Wear

Una vez configurado el reloj deberemos conectarlo a nuestro ordenador para instalar la APK, para ello deberemos:

1. Deberemos iniciar la consola de comandos y situarnos dentro de la carpeta de platform-tools del Sdk de Android donde estará ubicado el archivo adb que necesitaremos para realizar la instalación.
2. Deberemos introducir el siguiente código (recordar IP obtenida en el paso 3 de la configuración del reloj):
Adb connect "IP y puerto del paso 3 de la configuración del reloj"
3. Una vez conectado el reloj al ordenador podremos realizar la instalación mediante el siguiente código:
Adb install "ruta de nuestra APK"

Para realizar la instalación en nuestro teléfono Android, tendremos que habilitar la instalación de aplicaciones de orígenes desconocidos, ya que al no estar en la Play Store no se reconocerá como aplicación oficial, copiaremos la APK en el teléfono y la podremos instalar.

8.2. Manual de uso

Para comenzar a utilizar la aplicación se deberá empezar utilizando la aplicación para teléfono, esta aplicación no requiere de ningún permiso adicional, pero para el correcto funcionamiento del sistema se deberá descargar la siguiente aplicación de Android Wear:

<https://play.google.com/store/apps/details?id=com.google.android.wearable.app>

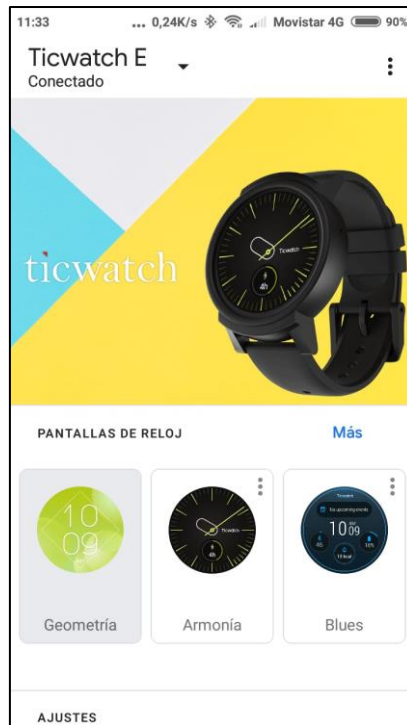


Ilustración 8.3 Aplicación Android Wear

Deberemos configurar el reloj en la aplicación de Android Wear para facilitar la sincronización de nuestra aplicación.

Una vez configurado el reloj, podremos comenzar a utilizar Refwatch, nada más iniciar la aplicación nos encontraremos con la siguiente pantalla de inicio.



Ilustración 8.4 Pantalla principal de la aplicación móvil

Desde esta pantalla podremos acceder a la configuración de partidos, a los partidos pendientes y a los partidos ya jugados.

Inicialmente deberemos crear un partido.



The screenshot shows a mobile application interface for creating a match. At the top, there is a back arrow and the title 'CREAR PARTIDO'. Below this, there are several input fields with labels and values: 'Duración del partido' with the value '90', 'Duración del descanso' with '15', 'Nº sustituciones' with '3', and 'Nº sustitutos' with '7'. There is also a 'Fecha del partido' field which is currently empty. Below these fields is a checkbox labeled 'Prorroga' which is unchecked. At the bottom of the form is a button labeled 'CONFIGURAR EQUIPOS'.

Ilustración 8.5 Pantalla de configuración de partidos

Para crear un partido deberemos rellenar todos los campos solicitados por el sistema, dependiendo de si seleccionamos prórroga o no, se habilitarán o deshabilitarán los campos relacionados con la prórroga. Una vez estén completados los campos seleccionaremos configurar equipos para proseguir con la configuración del partido.



The screenshot shows the 'CREAR PARTIDO' screen with the 'CONFIGURAR EQUIPOS' button selected. It displays two sections for team configuration. The first section is for the 'Equipo local' (Home Team), with a 'Nombre' (Name) field and a blue button labeled 'SELECCIONAR JUGADORES'. The second section is for the 'Equipo Visitante' (Away Team), also with a 'Nombre' field and a green button labeled 'SELECCIONAR JUGADORES'. At the bottom of the screen is a button labeled 'GUARDAR PARTIDO'.

Ilustración 8.6 Pantalla de configuración de equipos

En esta pantalla deberemos rellenar el nombre de los equipos y seleccionar los jugadores titulares y sustitutos de cada equipo, para ello deberemos seleccionar el botón “seleccionar jugadores” de cada equipo para elegir los jugadores.

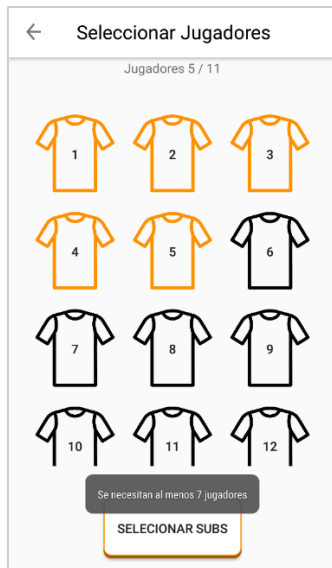


Ilustración 8.7 Pantalla de selección de jugadores

En esta pantalla deberemos elegir los titulares eligiendo al menos 7 jugadores y como mucho 11, tras elegir los jugadores titulares pasaremos a elegir los jugadores sustitutos, una vez elegidos los jugadores sustitutos volveremos a la creación de equipos, donde podremos seleccionar guardar partido para almacenar el partido y poder iniciarle desde la pantalla principal, seleccionando “Partidos pendientes”.



Ilustración 8.8 Partidos pendientes

Una vez en esta pantalla seleccionaremos el partido que queramos iniciar podremos ver la información del mismo, para empezar el partido deberemos pulsar en comenzar, y automáticamente se iniciará la aplicación del reloj mientras este esté conectado a nuestro teléfono.



Ilustración 8.9 Información sobre un partido

Tras iniciarse la aplicación en el reloj nos encontraremos con el partido ya configurado listo para ser iniciado.

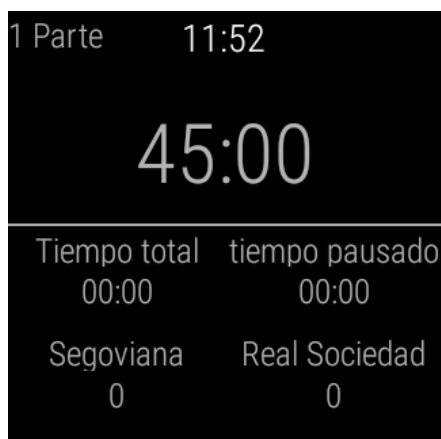


Ilustración 8.10 Pantalla principal de la aplicación Wear

Para iniciar el cronometro y para pausarle deberemos pulsar el cronometro principal de la aplicación.

Para añadir incidencias durante el partido deberemos deslizar la pantalla hacia arriba para abrir el menú de la aplicación.

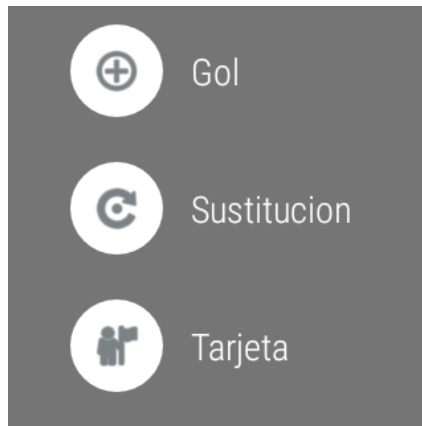


Ilustración 8.11 Menú de aplicación Wear

Para añadir goles seleccionaremos la opción de Gol en el menú y una vez en la pantalla de añadir goles podremos elegir el equipo que marcó el gol, si el gol ha sido de penalti o en propia puerta podremos seleccionar el tipo de gol y después elegir el jugador que marcó el gol.



Ilustración 8.12 Añadir gol

Para añadir tarjetas seleccionaremos la opción tarjeta en el menú de la aplicación, seleccionaremos el equipo del jugador sancionado y seleccionaremos al jugador, aquellos jugadores con una tarjeta aparecerán con una tarjeta amarilla sobre ellos, y los que tengan dos tarjetas aparecerán con una tarjeta roja.



Ilustración 8.13 Añadir tarjeta

Para añadir sustituciones deberemos elegir la opción sustitución en el menú de la aplicación, tras elegir esta opción deberemos especificar el equipo que quiere realizar la sustitución, una vez elegido el equipo deberemos elegir el jugador sustituido y el jugador sustituto.

Tras acabar el tiempo de la primera parte, comenzará el tiempo de descanso, y a partir de este momento el árbitro podrá seleccionar “Cambiar tiempo” en el menú de la aplicación, una vez seleccionada esta opción, los cronómetros se actualizarán para empezar un nuevo tiempo, el cronómetro principal volverá a tener el tiempo total del periodo, el cronómetro de pausa volverá a cero, y el cronómetro total restara el tiempo de pausa.

Una vez terminada la segunda parte si el partido ha terminado el árbitro podrá seleccionar terminar partido en el menú principal para enviar los resultados del partido al teléfono.

Una vez el partido ha sido enviado al teléfono, podremos ver los resultados del partido en la opción partidos jugados del menú principal de la aplicación del teléfono.

En esta pantalla podremos visualizar los resultados del partido con las distintas incidencias ocurridas a lo largo del partido.



Ilustración 8.14 Partido finalizado

Capítulo 9 Conclusiones

Al principio me planteaba realizar una aplicación Android, ya que desde la asignatura de Plataformas Software Móviles quería hacer algo relacionado, de ahí surgió la idea de hacer una aplicación wear, intentando diferenciarme de la clásica aplicación móvil. Al explorar las distintas ofertas de TFG, fue la propuesta de mi tutor Luis Ignacio Sebastián, la que encajaba perfectamente en aquello que yo buscaba de TFG.

Comenzando con este TFG empecé realizando una arquitectura modelo vista controlador, pero tras entrar a la empresa donde curse prácticas, aprendí una arquitectura distinta que me resulto muy llamativa y empecé a aprender nuevas tecnologías para poder aplicarlas en este TFG, como son la utilización de inyección de dependencias y las distintas conexiones con las API's de Android Wear.

La realización del proyecto ha servido para consolidar y ampliar lo aprendido a lo largo de la carrera, ya sea por la parte de análisis redactando este documento, como por la parte de desarrollo durante la creación de las aplicaciones.

A nivel personal, estoy bastante satisfecho con el resultado de este trabajo, ya que supone el cumplimiento de todas las funciones previstas, aunque al proyecto le faltan funcionalidades por mejorar, esto es debido al gran alcance del mismo, las funcionalidades sin desarrollar se plantean como mejoras del proyecto para futuras versiones.

9.1. Posibles ampliaciones

Durante el desarrollo del proyecto se han visto funcionalidades que resultarían interesantes en la aplicación pero que no se han podido llevar a cabo durante la versión actual del proyecto

- **Comentarios de incidencias por voz:** permitir al árbitro introducir comentarios por voz sobre las incidencias agilizaría la creación de incidencias y enriquecería el acta del partido con la posibilidad de saber la causa de alguna de las incidencias.
- **Vincular aplicación con google calendar:** esta vinculación permitiría guardar los distintos partidos en el calendario para avisarnos de las fechas y no olvidarnos de ningún partido, es una funcionalidad interesante para aquellos a los que les gusta organizar su tiempo.
- **Obtener partidos automáticamente de un servidor:** al recibir los partidos desde un servidor, el usuario no tendría que crear los partidos, tan solo tendría que seleccionar el partido para empezarle y la velocidad de uso de la aplicación sería aún más rápida. Con esta funcionalidad se podrían asignar a un árbitro los partidos de un torneo y se podría recuperar información adicional sobre los partidos, enriqueciendo aún más las actas postpartido.

- **Enviar actas de partidos a un servidor:** Al tener un servidor al que enviar las actas, la dirección del torneo ya tendría el acta nada más sincronizarlo con el servidor y facilitaría aún más la labor del árbitro.
- **Listado de jugadores por nombre:** actualmente el listado de los jugadores es mediante su dorsal, sin embargo si se dispusiera de la información recogida del servidor con los datos de los equipos se podría utilizar los nombres de los jugadores para de esta forma asociar a cada jugador con su dorsal.

Referencias bibliográficas

- Ajay. (17 de Mayo de 2017). *medium.com*. Obtenido de medium.com: <https://medium.com/@ajaysaini.official/building-database-with-room-persistence-library-ecf7d0b8f3e9>
- Campos, J. (2011). Obtenido de <https://itunes.apple.com/es/app/cronof%C3%BAAtbol/id466223662?mt=8>
- Cejas, F. (2014). *fernandocejas.com*. Obtenido de fernandocejas.com: <https://fernandocejas.com/2014/09/03/architecting-android-the-clean-way/>
- ComTech. (2018). *kantarworldpanel.com*. Obtenido de kantarworldpanel.com: <https://www.kantarworldpanel.com/global/smartphone-os-market-share/intro>
- Droiosapps. (27 de Marzo de 2016). *Apkpure*. Obtenido de Apkpure: <https://apkpure.com/cron%C3%B3metro-de-f%C3%BAAtbol/hw.apps.andro.com.cronometro>
- eleks.labs. (2014). *labs.eleks.com*. Obtenido de labs.eleks.com: <https://labs.eleks.com/2014/11/wearable-runtime-connectivity-architectures.html>
- Gonzalez, J. (10 de Noviembre de 2016). *http://bemobile.es*. Obtenido de <http://bemobile.es>: <http://bemobile.es/blog/2016/11/clean-architecture-the-bemobile-way/>
- Google. (15 de Junio de 2018). *developer.android.com*. Obtenido de developer.android.com: <https://developer.android.com/training/wearables/ui/layouts>
- Google. (s.f.). *Developer.android.com*. Obtenido de Developer.android.com: <https://developer.android.com/wear/>
- Goref. (2016). *Goref.co*. Obtenido de Goref.co: <http://goref.co/>
- pipper.publishing. (2018). Obtenido de <https://play.google.com/store/apps/details?id=com.pipperpublishing.soccerrefpro&hl=es>
- REFSIX Ltd. (2018). *refsix.com*. Obtenido de refsix.com: <https://refsix.com/>
- sergiocasero. (s.f.). *Geekytheory.com*. Obtenido de Geekytheory.com: <https://geekytheory.com/butterknife-sorprendentemente-sencillo-absolutamente-imprescindible>
- Sesma, M. (s.f.). *Paradigmadigital.com*. Obtenido de Paradigmadigital.com: <https://www.paradigmadigital.com/dev/inyeccion-dependencias-dagger-2/>
- Shekhar, A. (2016). *github.com*. Obtenido de github.com: <https://github.com/amitshekhariitbhu/Android-Debug-Database>

Subirats, J. (9 de Julio de 2014). *Elandroidelibre.espanol.com*. Obtenido de Elandroidelibre.espanol.com:
<https://elandroidelibre.espanol.com/2014/07/como-activar-las-opciones-de-desarrollo-en-android-wear.html>

Anexos

Anexo I: Atributos del ajuste de los puntos de función

Los 14 atributos que se contemplan en esta técnica y sus significados aparecen a continuación.

1. Comunicación de datos: Los datos usados en la aplicación se envían o reciben por teleproceso. Los posibles valores para este atributo son:

- **0** – La aplicación es un proceso por lotes puro.
- **1** – Proceso por lotes con impresión remota o entrada remota de datos.
- **2** – Proceso por lotes con impresión remota y entrada remota de datos.
- **3** – El TP es la interfaz para un proceso por lotes.
- **4** – La aplicación está basada en un TP interactivo, pero con un solo protocolo de comunicaciones.
- **5** – La aplicación está basada en un TP interactivo, pero con más de un protocolo de comunicaciones.

2. Funciones distribuidas: Funciones de datos o procesos distribuidas. Los posibles valores para este atributo son:

- **0** – La aplicación no tiene el objetivo de transferir datos o funciones procesadas entre dos sistemas.
- **1** – Datos preparados de la aplicación para su procesamiento por el usuario final sobre otro componente del sistema.
- **2** – La aplicación prepara los datos para procesarlos sobre otra máquina diferente (no usuario final).
- **3** – Proceso distribuido, en línea, con transferencia de datos en una única dirección.
- **4** – Como el anterior, pero con transferencia de datos en ambas direcciones.
- **5** – Las funciones de proceso se realizan dinámicamente sobre el componente del sistema más apropiado.

3. Prestaciones: Consideración en el diseño, instalación y mantenimiento de factores de rendimiento como el tiempo de respuesta, la capacidad de proceso, etc. Los posibles valores para este atributo son:

- **0** – No hay requerimientos especiales.
- **1** – Se establecen requerimientos para las prestaciones, pero sin tratamiento específico.
- **2** – Respuesta crítica del proceso en línea durante las horas punta. No hay especificaciones para la utilización de CPU.
- **3** – Respuesta crítica del proceso en línea durante los días laborables. No hay especificaciones para la utilización de CPU. Proceso afectado por aplicaciones de interfaz.
- **4** – Las tareas de análisis de las prestaciones se incluyen en la fase de diseño para establecer los requerimientos de usuario.
- **5** – Además, se emplearán herramientas específicas para el diseño que contemplen estas características.

4. Gran uso de la configuración: Cuando además de los objetivos de rendimiento se considera una gran utilización. El usuario ha de utilizar la aplicación en un entorno bastante cargado. Los posibles valores para este atributo son:

- **0-3** – Típica aplicación sobre máquina de producción, sin restricciones de operación declaradas.
- **4** – Las restricciones de operación declaradas requieren imperativos especiales sobre la aplicación en el procesador central.
- **5** – Además, existen imperativos especiales sobre la aplicación en componentes distribuidos del sistema.

5. Velocidad de las transacciones: Número alto de transacciones por unidad de tiempo que influyen en el diseño, instalación y posterior mantenimiento. Los posibles valores para este atributo son:

- **0** – Las transacciones no están afectadas por picos de tráfico.
- **1** – 10% de transacciones afectadas por los picos de tráfico.
- **2** – 50% de transacciones afectadas por los picos de tráfico.
- **3** – 100% de transacciones afectadas por los picos de tráfico.
- **4** – Se incluyen tareas de análisis para las funciones en la fase de diseño para lograr los altos índices de función declarados por el usuario en los requerimientos de la aplicación o acuerdos de nivel de servicio (SLA).
- **5** – Además, se utilizan herramientas de análisis para las prestaciones en las fases de diseño, desarrollo y / o instalación para lograr los altos índices de función declarados por el usuario en los requerimientos de la aplicación o acuerdos de nivel de servicio (SLA).

6. Entrada de datos en línea: La toma de datos de la aplicación se realiza en línea. Los posibles valores para este atributo son:

- **0** – Todas las transacciones son tratadas por lotes.
- **1** – Entre el 1 y el 7% de las funciones son entradas interactivas de datos.
- **2** – Entre el 8 y el 15% de las funciones son entradas interactivas de datos.
- **3** – Entre el 16 y el 23% de las funciones son entradas interactivas de datos.
- **4** – Entre el 24 y el 30% de las funciones son entradas interactivas de datos.
- **5** – Más del 30% de las funciones son entradas interactivas de datos.

7. Diseño para la eficiencia del usuario final: Se incluyen tareas de diseño para consideraciones especiales del usuario en la Fase de Diseño para atender los requerimientos del usuario, por ejemplo:

1. Ayuda de navegación.
2. Menús.
3. Ayuda en línea.
4. Movimiento automático del cursor.
5. Scrolling.
6. Impresión remota.
7. Teclas de función preestablecidas.
8. Procesos por lotes lanzados desde transacciones en línea.
9. Selección de datos con el cursor.
10. Gran uso de facilidades en el monitor (colores, textos resaltados, etc.).

11. Copia impresa de las transacciones en línea.
12. Ratón.
13. Windows.
14. Pantallas reducidas.
15. Bilingüismo.
16. Multilingüismo.

Los posibles valores para este atributo son:

- **0** – No se han declarado ninguno de los anteriores requerimientos especiales de usuario.
- **1** – De 1 a 3 de los requerimientos de la lista.
- **2** – 4 ó 5 requerimientos de la lista.
- **3** – Más de 6 requerimientos de la lista.
- **4** – Se incluyen en la fase de diseño tareas de diseño para consideraciones de factores humanos para lograr los requerimientos de usuario declarados.
- **5** – Además, se usan herramientas especiales o prototipos para suscitar la eficiencia del usuario final.

8. Actualización de datos en línea: Los datos internos se actualizan mediante transacciones en línea. Los posibles valores para este atributo son:

- **0** – Ninguna.
- **1-2** – Actualización en línea de ficheros de control.
- **3** – Actualización en línea de ficheros importantes internos.
- **4** – También, se considera esencial la protección contra pérdida de información.
- **5** – Además, grandes volúmenes implican consideraciones de coste en el proceso de recuperación.

9. Complejidad del proceso lógico interno de la aplicación: Se considera complejo cuando hay muchas interacciones, puntos de decisión o gran número de ecuaciones lógicas o matemáticas. ¿Cuál de las siguientes características tienen aplicación para la aplicación?

1. Extensiones de proceso lógicas.
2. Extensiones de proceso matemáticas.
3. Muchos procesos de excepción, muchas funciones incompletas y muchas iteraciones de funciones.
4. Procesos sensibles de control y / o seguridad.
5. Procesos complejos de manejo de múltiples posibilidades de Entrada / Salida (por ejemplo: multimedia, independencia de dispositivos,...).

Los posibles valores para este atributo son:

- **0** – Ninguno de los anteriores es aplicable.
- **1** – Es aplicable uno de los anteriores.

- 2 – Son aplicables dos de los anteriores.
- 3 – Son aplicables 3 de los anteriores.
- 4 – Son aplicables 4 de los anteriores.
- 5 – Todos ellos son aplicables.

10. **Reusabilidad del código** por otras aplicaciones. Los posibles valores para este atributo son:

- 0 – No hay que reutilizar el código.
- 1 – Se emplea código reusable dentro de la aplicación.
- 2 – Menos del 10% de la aplicación se considera reusable.
- 3 – El 10% o más de la aplicación se considera reusable.
- 4 – La aplicación está específicamente preparada y documentada para facilitar la reutilización y se adapta sobre código fuente.
- 5 – La aplicación está específicamente preparada y documentada para facilitar la reutilización y, además, se adapta sobre parámetros.

11. **Facilidad de instalación:** Durante el desarrollo se consideran factores que facilitan la ulterior conversión e instalación. Los posibles valores para este atributo son:

- 0 – El usuario no ha declarado consideraciones especiales para instalación y conversión.
- 1 – El usuario no ha declarado consideraciones especiales para instalación y conversión, pero se requiere un set especial para la instalación.
- 2 – El usuario ha declarado consideraciones especiales para la conversión e instalación y se requieren guías probadas de conversión e instalación.
- 3 – El usuario ha declarado consideraciones especiales para la conversión e instalación y se requieren guías probadas de conversión e instalación y se considera importante el impacto.
- 4 – El usuario ha declarado consideraciones especiales para la conversión e instalación y se requieren guías probadas de conversión e instalación y, además, se facilitan herramientas probadas para la conversión e instalación.
- 5 – El usuario ha declarado consideraciones especiales para la conversión e instalación y se requieren guías probadas de conversión e instalación, considerándose importante el impacto. Además, se facilitan herramientas probadas para la conversión e instalación.

12. **Facilidad de operación:** Se han tenido en cuenta factores de operatividad. Se han considerado procedimientos de arranque, de copia de respaldo y de recuperación. Los posibles valores para este atributo son:

- 0 – No hay consideraciones especiales de operación.
- 1-2 – Se requieren procesos específicos de arranque, back-up y recuperación debidamente probados.
- 3-4 – Además, la aplicación debe minimizar las necesidades de operaciones manuales, como manejo de papeles o montaje de cintas.
- 5 – La aplicación debe diseñarse para una operación totalmente automática.

13. **Localizaciones múltiples:** La aplicación se diseña para ser utilizada en diversas instalaciones y por organizaciones. El valor para este atributo será la suma de los aplicables:

- 0 – No hay requerimientos de usuario para más de un lugar.

- **1** – Se consideran múltiples instalaciones pero con idéntica configuración (tanto hardware como software).
- **2** – Se consideran múltiples instalaciones pero con similar configuración (tanto hardware como software).
- **3** – Se consideran múltiples instalaciones pero con diferente configuración (tanto hardware como software).
- Se añadirá 1 punto por cada una de las siguientes consideraciones:
- Se proporcionará documentación y plan de soporte debidamente probados para soportar la aplicación en múltiples sitios.
- Los lugares están en diferentes países.

14. Facilidad de cambios: Se han tenido en cuenta criterios que facilitarán el posterior mantenimiento. El valor para este atributo será la suma de los aplicables:

- **0** – No hay requerimientos especiales de diseño para minimizar o facilitar los cambios.
- **1** – Se preverá una flexible capacidad de peticiones para modificaciones sencillas.
- **2** – Se preverá una flexible capacidad de peticiones para modificaciones medias.
- **3** – Se preverá una flexible capacidad de peticiones para modificaciones complejas.
- Se añadirán 1 ó 2 puntos dependiendo de que los datos de control significativos se guarden en tablas mantenidas por el usuario mediante procesos interactivos en línea:
- 1 para actualización diferida.
- 2 para actualización inmediata.

Anexo II: Especificación de casos de uso

Aplicación móvil

CU-01	Configurar partido	
Versión	1.0	
Requisitos asociados	CU-02, CU-03	
Actores	Usuario	
Descripción	El usuario crea un partido	
Precondiciones	-	
Secuencia normal	Paso	Acción
	P1	El usuario solicita la creación de un nuevo partido.
	P2	El sistema presenta los campos necesarios para la creación.
	P3	El usuario rellena los datos ofrecidos por el sistema.
	P4	El sistema valida la información introducida.
	P5	El caso de uso finaliza correctamente.
Postcondiciones	Los datos introducidos para la creación del partido son correctos.	
Excepciones	Paso	Acción
	P4	El sistema detecta que la información introducida no es válida.
Frecuencia	Diaria	
Importancia	Alta. La creación de partidos es necesaria para el funcionamiento básico de la aplicación.	
Comentarios	Cada partido será único identificado por el timestamp del momento de su creación.	

Tabla 61 CU-01 Configurar partido

CU-02	Configurar equipos	
Versión	1.0	
Requisitos asociados	CU-01	
Actores	Usuario	
Descripción	El usuario configura los jugadores de un equipo.	
Precondiciones	Se deben de haber elegido previamente las opciones del partido.	
Secuencia normal	Paso	Acción
	P1	El usuario solicita configurar los jugadores de un equipo.
	P2	El sistema muestra un listado con todos los jugadores.
	P3	El usuario elige jugador a jugador los titulares.
	P4	El sistema marca como seleccionados los jugadores del equipo cambiando el color de la selección.
	P5	El sistema actualiza el contador de jugadores elegidos.
	P6	El usuario solicita seleccionar substitutos.
	P7	El sistema muestra un listado con todos los jugadores no seleccionados como titulares.
	P8	El usuario elige jugador a jugador los substitutos.
	P9	El sistema marca como seleccionados los jugadores del equipo cambiando el color de la selección.
	P10	El sistema actualiza el contador de jugadores elegidos.
	P11	El usuario solicita guardar la configuración del equipo.
P12	El caso de uso finaliza correctamente.	
Postcondiciones	La configuración del equipo queda guardada correctamente.	
Excepciones	Paso	Acción
	P3	Si al elegir un jugador este ya estaba seleccionado el jugador se desmarca.
	P3	Si al elegir un jugador ya se ha alcanzado el número máximo de jugadores no se podrá añadir dicho jugador al menos que se desmarque otro jugador previamente.
	P6	Si se seleccionan menos de 7 jugadores se mostrará una notificación indicando que se necesitan al menos 7 jugadores
	P8	Si al elegir un jugador este ya estaba seleccionado el jugador se desmarca.
	P8	Si al elegir un jugador ya se ha alcanzado el número máximo de jugadores no se podrá añadir dicho jugador al menos que se desmarque otro jugador previamente.
Frecuencia	Diaria	
Importancia	Alta. Función necesaria para las realizar sustituciones durante el partido.	
Comentarios	Tras completarse el caso de uso el botón de configuración del equipo correspondiente cambiará de color para indicar que se ha completado su configuración	

Tabla 62 CU-02 Configurar equipos

CU-03	Guardar partido pendiente	
Versión	1.0	
Requisitos asociados	CU-01	
Actores	Usuario	
Descripción	Guarda el partido en la base de datos.	
Precondiciones	Datos correctamente validados.	
Secuencia normal	Paso	Acción
	P1	El usuario solicita guardar el partido
	P2	El sistema comprueba que todos los datos introducidos son válidos y que los equipos están configurados correctamente.
	P3	El sistema guarda el partido en la base de datos local.
	P4	El sistema notifica que el partido se ha guardado correctamente.
	P5	El sistema vuelve a la pantalla principal de la aplicación.
	P6	El caso de uso finaliza correctamente.
Postcondiciones	El partido se ha guardado correctamente	
Excepciones	Paso	Acción
	P2	El sistema muestra una notificación indicando que falta por introducir datos para guardar el partido
	P3	El sistema muestra una notificación indicando que el partido no se ha podido guardar en la base de datos correctamente.
Frecuencia	Diaria	
Importancia	Alta. Función necesaria para crear partidos.	
Comentarios		

Tabla 63 CU-03 Guardar partido pendiente

CU-04	Mostrar partidos pendientes	
Versión	1.0	
Requisitos asociados	CU-05	
Actores	Usuario	
Descripción	Muestra un listado de los partidos pendientes	
Precondiciones	-	
Secuencia normal	Paso	Acción
	P1	El usuario solicita mostrar el listado de partidos pendientes.
	P2	El sistema obtiene el listado de partidos pendientes de la base de datos local.
	P3	El sistema muestra el listado de los partidos pendientes.
	P4	El caso de uso finaliza correctamente.
Postcondiciones	Se muestran los partidos pendientes.	
Excepciones	Paso	Acción
	P2	Si no existen partidos en la base de datos no se mostrará nada.
Frecuencia	Diaria	
Importancia	Alta. Función necesaria para iniciar un partido	
Comentarios		

Tabla 64 CU-04 Mostrar partidos pendientes

CU-05	Mostrar partido pendiente	
Versión	1.0	
Requisitos asociados	CU-04, CU-06, CU-07	
Actores	Usuario	
Descripción	Muestra un partido pendiente.	
Precondiciones	Se debe seleccionar un partido del listado de partidos.	
Secuencia normal	Paso	Acción
	P1	El usuario solicita mostrar un partido.
	P2	El sistema obtiene la información del partido correspondiente.
	P3	El sistema muestra el partido pendiente.
	P4	El caso de uso finaliza correctamente.
Postcondiciones	Se muestra el partido pendiente correspondiente.	
Excepciones	Paso	Acción
	P2	Si el sistema no puede obtener la información del partido deseado se mostrara una notificación de error
Frecuencia	Diaria	
Importancia	Media. Es una funcionalidad importante pero no es imprescindible para el funcionamiento de la aplicación.	
Comentarios		

Tabla 65CU-05 Mostrar partido pendiente

CU-06	Borrar partido pendiente	
Versión	1.0	
Requisitos asociados	CU-05	
Actores	Usuario	
Descripción	Borra el partido pendiente seleccionado.	
Precondiciones	Debe existir un partido pendiente.	
Secuencia normal	Paso	Acción
	P1	El usuario solicita borrar un partido.
	P2	El sistema solicita confirmación para borrar el partido.
	P3	El usuario acepta borrar el partido.
	P4	El sistema borra el partido de la base de datos.
	P5	El sistema muestra una notificación mostrando que el partido ha sido borrado de la base de datos.
	P6	El sistema vuelve a la pantalla anterior.
	P7	El caso de uso finaliza correctamente.
Postcondiciones	El partido ha sido borrado	
Excepciones	Paso	Acción
	P3	Si el usuario cancela la eliminación del partido no ocurrirá nada.
	P4	Si el sistema no puede borrar el partido de la base de datos se mostrara una notificación del error correspondiente.
Frecuencia	Esporádica	
Importancia	Media/Baja. Importante cuando hay gran cantidad de partidos o tras equivocarse en la creación de un partido con datos erróneos.	
Comentarios		

Tabla 66 CU-06 Borrar partido pendiente

CU-07	Empezar partido pendiente	
Versión	1.0	
Requisitos asociados	CU-05	
Actores	Usuario, Bluetooth	
Descripción	Manda los datos del partido mediante bluetooth.	
Precondiciones	Debe existir un partido.	
Secuencia normal	Paso	Acción
	P1	El usuario solicita enviar un partido.
	P2	El sistema solicita comprueba la conexión bluetooth.
	P3	El sistema manda el partido por bluetooth.
	P4	El sistema muestra una notificación de partido enviado correctamente.
P5	El caso de uso finaliza correctamente.	
Postcondiciones	El partido ha sido enviado.	
Excepciones	Paso	Acción
	P2	Si el sistema no tiene la conexión bluetooth habilitada se muestra un mensaje de error.
	P3	Si no se puede mandar el mensaje se muestra una notificación del error.
Frecuencia	Diaria	
Importancia	Alta. Es necesario realizar envíos de partidos al reloj.	
Comentarios	Para conectarse con el teléfono deberá estar el reloj sincronizado mediante Wear OS al Android wear para hacer uso de las API's del mismo.	

Tabla 67 CU-07 Empezar partido pendiente

CU-08	Recibir partido terminado	
Versión	1.0	
Requisitos asociados		
Actores	Bluetooth	
Descripción	Guarda un partido terminado.	
Precondiciones	Debe estar el bluetooth habilitado.	
Secuencia normal	Paso	Acción
	P1	El bluetooth recibe un partido.
	P2	El sistema valida los datos del partido.
	P3	El sistema guarda el partido en la base de datos
	P4	El caso de uso finaliza correctamente.
Postcondiciones	El partido terminado queda guardado en la base de datos local.	
Excepciones	Paso	Acción
	P3	Si el sistema no puede almacenar el partido en la base de datos local se mostrara una notificación de la incidencia.
Frecuencia	Diaria	
Importancia	Alta. Funcionalidad necesaria para recoger los datos del partido.	
Comentarios	Para conectarse con el teléfono deberá estar el reloj sincronizado mediante Wear OS al Android wear para hacer uso de las API's del mismo.	

Tabla 68 CU-08 Recibir partido terminado

CU-09	Mostrar partidos terminados	
Versión	1.0	
Requisitos asociados	CU-10	
Actores	Usuario	
Descripción	Muestra un listado de los partidos pendientes	
Precondiciones	-	
Secuencia normal	Paso	Acción
	P1	El usuario solicita mostrar el listado de partidos terminados.
	P2	El sistema obtiene el listado de partidos terminados de la base de datos local.
	P3	El sistema muestra el listado de los partidos terminados.
	P4	El caso de uso finaliza correctamente.
Postcondiciones	Se muestran los partidos terminados.	
Excepciones	Paso	Acción
	P2	Si no existen partidos terminados en la base de datos local no se muestra nada.
Frecuencia	Diaria	
Importancia	Alta. Función necesaria para iniciar un partido	
Comentarios		

Tabla 69 CU-09 Mostrar partidos terminados

CU-10	Mostrar partido terminado	
Versión	1.0	
Requisitos asociados	CU-09, CU-11	
Actores	Usuario	
Descripción	Muestra un partido pendiente.	
Precondiciones	Se debe seleccionar un partido del listado de partidos.	
Secuencia normal	Paso	Acción
	P1	El usuario solicita mostrar un partido.
	P2	El sistema obtiene la información del partido correspondiente.
	P3	El sistema muestra el partido terminado.
	P4	El caso de uso finaliza correctamente.
Postcondiciones	Se muestra el partido pendiente correspondiente.	
Excepciones	Paso	Acción
	P2	Si el sistema no puede obtener la información del partido deseado se mostrara una notificación de error
Frecuencia	Diaria	
Importancia	Alta. Es necesario ver la información del partido tras su finalización.	
Comentarios		

Tabla 70 CU-10 Mostrar partido terminado

CU-11	Borrar partido terminado	
Versión	1.0	
Requisitos asociados	CU-10	
Actores	Usuario	
Descripción	Borra el partido pendiente seleccionado.	
Precondiciones	Debe existir un partido pendiente.	
Secuencia normal	Paso	Acción
	P1	El usuario solicita borrar un partido.
	P2	El sistema solicita confirmación para borrar el partido.
	P3	El usuario acepta borrar el partido.
	P4	El sistema borra el partido de la base de datos.
	P5	El sistema muestra una notificación mostrando que el partido ha sido borrado de la base de datos.
	P6	El sistema vuelve a la pantalla anterior.
	P7	El caso de uso finaliza correctamente.
Postcondiciones	El partido ha sido borrado	
Excepciones	Paso	Acción
	P3	Si el usuario cancela la eliminación del partido no ocurrirá nada.
	P4	Si el sistema no puede borrar el partido de la base de datos se mostrara una notificación del error correspondiente.
Frecuencia	Esporádica	
Importancia	Media/Baja. Importante cuando hay gran cantidad de partidos o tras equivocarse en la creación de un partido con datos erróneos.	
Comentarios		

Tabla 71 CU-11 Borrar partido terminado

CU-12	Recibir partido	
Versión	1.0	
Requisitos asociados		
Actores	Bluetooth	
Descripción	Recibe los datos de un partido	
Precondiciones	Conexión bluetooth habilitada	
Secuencia normal	Paso	Acción
	P1	El Bluetooth recibe un partido.
	P2	El sistema comprueba los datos del partido
	P3	El sistema inicializa los campos del cronometro con los datos del partido.
	P4	Finaliza el caso de uso correctamente
Postcondiciones	El partido tiene los campos inicializados.	
Excepciones	Paso	Acción
	P2	Si los datos del partido están incompletos manda una notificación sobre el error.
Frecuencia	Diaria	
Importancia	Alta. Es una función necesaria para empezar el partido.	
Comentarios	Para conectarse con el teléfono deberá estar el reloj sincronizado mediante Wear OS al Android wear para hacer uso de las API's del mismo.	

Tabla 72 CU-12 Recibir partido

CU-13	Terminar partido	
Versión	1.0	
Requisitos asociados	CU-20	
Actores	Usuario, Bluetooth, Reloj	
Descripción	Manda el partido por bluetooth	
Precondiciones	El partido debe de haber terminado.	
Secuencia normal	Paso	Acción
	P1	El usuario solicita terminar el partido.
	P2	El sistema comprueba que los datos del partido son correctos.
	P3	El sistema comprueba que el partido haya terminado previamente.
	P4	El sistema comprueba la conexión bluetooth.
	P5	El sistema envía el partido por bluetooth.
	P6	El sistema notifica que el partido se ha enviado con éxito.
	P7	El sistema cierra la aplicación wear.
	P8	El caso de uso finaliza correctamente.
Postcondiciones	El partido ha sido enviado.	
Excepciones	Paso	Acción
	P2	Si los datos son incorrectos el sistema mostrara una notificación con el error correspondiente.
	P3	Si el partido no ha terminado no se continuará con el envío del partido.
	P4	Si no se puede enviar el partido por bluetooth el sistema mostrara una notificación con el error correspondiente.
Frecuencia	Diaria	
Importancia	Alta. Es necesario el envío de partidos terminados al reloj	
Comentarios	Para conectarse con el teléfono deberá estar el reloj sincronizado mediante Wear OS al Android wear para hacer uso de las API's del mismo.	

Tabla 73 CU-13 Terminar partido

CU-14	Iniciar/pausar cronómetro	
Versión	1.0	
Requisitos asociados	CU-19	
Actores	Usuario, Reloj	
Descripción	Inicia o pausa el cronómetro.	
Precondiciones	El partido debe de estar inicializado.	
Secuencia normal	Paso	Acción
	P1	El usuario solicita iniciar/pausar cronometro
	P2	El sistema comprueba si el cronometro principal esta iniciado, en pausa o es su primer inicio.
	P3.1	Si el cronómetro esta pausado el sistema lo inicia y pausa el cronómetro de pausa.
	P3.2	Si el cronómetro esta iniciado el sistema lo pausa e inicia el cronómetro de pausa.
	P3.3	Si era el primer inicio del cronometro principal inicia el cronometro total
P4	El caso de uso termina correctamente.	
Postcondiciones	El cronometro se activa/desactiva.	
Excepciones	Paso	Acción
	P1	Si el cronómetro principal está a 0 el sistema no hace nada porque el periodo ya ha terminado
Frecuencia	Constante.	
Importancia	Vital. Función principal de la aplicación para controlar los cronómetros	
Comentarios		

Tabla 74 CU-13 Iniciar/pausar cronómetro

CU-15	Cambiar periodo	
Versión	1.0	
Requisitos asociados	CU-20	
Actores	Usuario, Reloj	
Descripción	Cambia el periodo del partido	
Precondiciones		
Secuencia normal	Paso	Acción
	P1	El usuario solicita cambiar de periodo.
	P2	El sistema comprueba el tiempo del cronómetro principal.
	P3	El sistema comprueba que periodo es y si puede haber prorroga.
	P4	El sistema inicializa el cronómetro principal con el tiempo correspondiente al siguiente periodo.
	P5	El sistema modifica el cronómetro total, eliminando el tiempo de pausa.
	P6	El sistema inicializa el cronómetro de pausa a 0 para el nuevo periodo.
	P7	El sistema modifica el texto del periodo actual indicando el nuevo periodo.
	P8	El caso de uso finaliza correctamente.
Postcondiciones	El periodo ha sido cambiado.	
Excepciones	Paso	Acción
	P2	Si el cronometro principal no ha llegado a 0, el sistema mostrara una notificación de que el periodo aún no ha terminado.
	P3	Si el periodo actual es la segunda parte y no hay prorroga, el partido ha terminado y no se cambiará de periodo
	P3	Si el periodo actual es la segunda parte y hay prorroga, el partido terminará si el resultado es distinto de empate.
Frecuencia	Diaria	
Importancia	Vital. Función principal de la aplicación para controlar los cronómetros	
Comentarios		

Tabla 75 CU-15 Cambiar periodo

CU-16	Añadir gol	
Versión	1.0	
Requisitos asociados		
Actores	Usuario	
Descripción	Añade una incidencia de tipo gol	
Precondiciones	El periodo no puede haber terminado	
Secuencia normal	Paso	Acción
	P1	El usuario solicita añadir un gol.
	P2	El sistema muestra un listado con los jugadores.
	P3	El usuario elige la pestaña del equipo que metió el gol.
	P4	El usuario selecciona el tipo de gol.
	P5	El usuario selecciona el jugador que marcó el gol.
	P6	El sistema añade la incidencia.
P7	El caso de uso termina correctamente.	
Postcondiciones	Se ha añadido una nueva incidencia de tipo gol.	
Excepciones		
Frecuencia	Diaria	
Importancia	Alta. Es importante la gestión de las incidencias.	
Comentarios	Un gol puede ser de tipo: Gol, Autogol, Penal.	

Tabla 76 CU-16 Añadir gol

CU-17	Añadir substitución	
Versión	1.0	
Requisitos asociados		
Actores	Usuario	
Descripción	Añade una incidencia de tipo substitución	
Precondiciones	El periodo no puede haber terminado	
Secuencia normal	Paso	Acción
	P1	El usuario solicita añadir una substitución.
	P2	El sistema permite elegir entre los dos equipos.
	P3	El usuario elige el equipo que va a realizar la substitución.
	P4	El usuario selecciona el jugador que abandonará el terreno.
	P5	El usuario selecciona el jugador que entrara al terreno de juego.
	P6	El sistema añade la incidencia.
P7	El caso de uso termina correctamente.	
Postcondiciones	Se ha añadido una nueva incidencia de tipo substitución.	
Excepciones	Paso	Acción
	P1	Si el equipo seleccionado no tiene jugadores substitutos, el sistema no hace nada y vuelve a la pantalla anterior.
Frecuencia	Diaria	
Importancia	Alta. Es importante la gestión de las incidencias.	
Comentarios		

Tabla 77 CU-17 Añadir substitución

CU-18	Añadir tarjeta	
Versión	1.0	
Requisitos asociados		
Actores	Usuario	
Descripción	Añade una incidencia de tipo tarjeta	
Precondiciones	El periodo no puede haber terminado	
Secuencia normal	Paso	Acción
	P1	El usuario solicita añadir una tarjeta.
	P2	El sistema muestra un listado con los jugadores.
	P3	El usuario elige la pestaña del equipo que cometió la infracción.
	P4	El usuario selecciona el jugador pendiente de amonestación.
	P5	El sistema añade la incidencia.
	P6	El caso de uso termina correctamente.
Postcondiciones	Se ha añadido una nueva incidencia de tipo tarjeta.	
Excepciones	Paso	Acción
	P2	Si algún jugador del listado tiene ya alguna tarjeta se mostrara una tarjeta al lado del jugador
	P4	Si el jugador ya tiene una tarjeta roja no se le podrá poner más infracciones.
Frecuencia	Diaria	
Importancia	Alta. Es importante la gestión de las incidencias.	
Comentarios		

Tabla 78 CU-18 Añadir tarjeta

CU-19	Actualizar cronómetros	
Versión	1.0	
Requisitos asociados	CU-14	
Actores	Reloj	
Descripción	Actualiza los cronómetros	
Precondiciones	El cronometro debe de estar iniciado	
Secuencia normal	Paso	Acción
	P1	El reloj recibe un evento de tiempo.
	P2	El sistema actualiza el tiempo de los cronómetros.
	P3	El caso de uso finaliza correctamente.
Postcondiciones	El tiempo de los cronómetros se actualiza.	
Excepciones		
Frecuencia	Constante. En cada segundo durante la ejecución de la aplicación wear.	
Importancia	Imprescindible. Funcionalidad esencial de la aplicación wear.	
Comentarios		

Tabla 79 CU-19 Actualizar cronómetros

CU-20	Finalizar cronómetros	
Versión	1.0	
Requisitos asociados	CU-13, CU-15	
Actores	Reloj	
Descripción	Finaliza los cronómetros	
Precondiciones	El cronometro debe de estar iniciado	
Secuencia normal	Paso	Acción
	P1	El reloj recibe un evento finalización de tiempo.
	P2	El sistema finaliza el cronómetro.
	P3	El caso de uso finaliza correctamente.
Postcondiciones	El tiempo de los cronómetros ha finalizado.	
Excepciones		
Frecuencia	Constante. Cada vez que termina un cronómetro.	
Importancia	Imprescindible. Funcionalidad esencial de la aplicación wear.	
Comentarios		

Tabla 80 CU-20 Finalizar cronómetros