



## PitchPutt ScoreCard

**Autor/Estudiante: Juan Luis Canales Canales**

*Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles*

**Profesor Consultor: Eduard Martin Lineros**

Fecha de entrega : 02/01/2019



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

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	PitchPutt ScoreCard
<b>Nombre del autor:</b>	Juan Luis Canales Canales
<b>Nombre del consultor:</b>	Eduard Martin Lineros
<b>Fecha de entrega (mm/aaaa):</b>	01/2019
<b>Titulación:</b>	<i>Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles</i>

### Resumen del Trabajo (máximo 250 palabras):

Aplicación Android enfocada al golf y pitch and putt, que permitirá a los usuarios no solo registrar sus tarjetas que actualmente pueden estar anotando a mano y lápiz, sino que se propone como una tool que le pueda dar más información sobre el campo y hoyo que está jugando, además de los metros, handicap del hoyo, par del hoyo la app calculará en todo momento según la modalidad del juego escogida la puntuación actual, y cual de los dos jugadores va ganando, así como la distancia hasta green del hoyo en curso. Adicionalmente permitirá guardar partidas, y registrar usuarios.

Es una aplicación enfocada únicamente al uso desde móviles, no se contempla el uso desde un PC o una tablet, ya que está enfocada a llevarla por el campo mientras se desarrolla el juego como una tool/adviser del jugador. Y además se plantea para que pueda funcionar aunque no haya cobertura a internet, ya que muchos campos se encuentran en zonas con no muy buena cobertura.

### Abstract (in English, 250 words or less):

This is an Android app focused to golf & pitch and putt, that will allow to users not only register the score cards (currently they can do it manually with a pencil), but also is proposing as a tool that could bring them more information about the golf course and the current hole playing. Apart of meters, hole handicap, par, this app will calculate based in the mode of the play choosen the current score, whose of the players is wining and the distance to the green of the hole. Aditionally will allow save the score cards / matches and register users.

It is an app focused only to be used with smart phones, due the app is useful just if you bring it during the match as a tool/adviser for the player. It is also planned to run even without data coverage to internet, because we know lot of golf courses are in zones without good data coverage

### Palabras clave (entre 4 y 8):

Golf Pitch Putt ScoreCard pitchandputt

## Índice

1	Introducción.....	1
1.1	Contexto y justificación del Trabajo .....	1
1.2	Objetivos del Trabajo .....	4
1.3	Enfoque y método seguido .....	5
1.4	Planificación del Trabajo .....	6
1.5	Breve resumen de productos obtenidos.....	8
1.6	Breve descripción de los otros capítulos de la memoria .....	8
2	Fase de Requerimientos .....	10
2.1	Usuarios.....	10
2.2	Indagación de requerimientos .....	11
2.3	Contexto de uso y Customer Journey Map .....	12
2.4	Escenarios de uso.....	14
2.5	Análisis de la competencia y valor añadido .....	16
2.6	Funcionalidades a incluir en la app.....	17
3	Fase de Diseño .....	23
3.1	Estrategia de contenido .....	23
3.2	Wireframes.....	23
3.3	Navegación .....	31
3.4	Prototipo .....	33
3.5	Diseño técnico .....	33
4	Evaluación y correcciones del diseño.....	41
5	Implementación.....	42
5.1	Modelo MVP.....	42
5.2	Estructura del proyecto .....	43
5.3	Views .....	45
5.4	Models.....	45
5.5	Presenters .....	46
5.6	Entities.....	46
5.7	Firebase .....	46
6	Valoración económica del trabajo .....	51
6.1	Gastos y Beneficios .....	51
6.2	Viabilidad del producto.....	51
7	Conclusiones.....	52
8	Glosario .....	54
9	Derechos de imagen.....	55
	CC0 License.....	55
10	Bibliografía.....	56
11	Anexos.....	57

## Lista de figuras

<u><a href="#">1.1 Disponibilidad</a></u> .....	<u><a href="#">5</a></u>
<u><a href="#">1.2 Planificación PEC2: Requerimientos &amp; Diseño</a></u> .....	<u><a href="#">5</a></u>
<u><a href="#">1.3 Planificación PEC3: Implementación</a></u> .....	<u><a href="#">5</a></u>
<u><a href="#">1.4 Planificación PEC4: Entrega</a></u> .....	<u><a href="#">6</a></u>
<u><a href="#">2.1 Tarjeta de puntuación</a></u> .....	<u><a href="#">9</a></u>
<u><a href="#">2.2 Ficha de usuario senior</a></u> .....	<u><a href="#">13</a></u>
<u><a href="#">2.3 Ficha usuario mediana edad</a></u> .....	<u><a href="#">13</a></u>
<u><a href="#">2.4 Encuesta funcionalidades</a></u> .....	<u><a href="#">15</a></u>
<u><a href="#">2.5 Contexto de uso, funcionalidades y condicionantes</a></u> .....	<u><a href="#">16</a></u>
<u><a href="#">2.6 Mapa de experiencia de usuario</a></u> .....	<u><a href="#">17</a></u>
<u><a href="#">2.7 ScoreCard: Pros &amp; Cons</a></u> .....	<u><a href="#">19</a></u>
<u><a href="#">2.8 Hole 19: Pros &amp; Cons</a></u> .....	<u><a href="#">20</a></u>
<u><a href="#">3.1 Diagrama de navegación</a></u> .....	<u><a href="#">34</a></u>
<u><a href="#">3.2 Diagrama de casos de uso</a></u> .....	<u><a href="#">36</a></u>
<u><a href="#">3.3 Tiers arquitectura aplicación</a></u> .....	<u><a href="#">37</a></u>
<u><a href="#">3.4 Conexiones aplicación</a></u> .....	<u><a href="#">38</a></u>
<u><a href="#">3.5 Diagrama Entidad - Relacion (I)</a></u> .....	<u><a href="#">40</a></u>
<u><a href="#">3.6 Diagrama Entidad - Relación (II)</a></u> .....	<u><a href="#">41</a></u>
<u><a href="#">3.7 Diagrama de clases activities</a></u> .....	<u><a href="#">41</a></u>
<u><a href="#">3.8 Diagrama de clases entidades</a></u> .....	<u><a href="#">42</a></u>

# 1 Introducción

## 1.1 Contexto y justificación del Trabajo

El trabajo/producto será enfocado al área de deportes, y más concretamente al juego del golf/pitch&putt (modalidad del golf más económica que se juega en campos más cortos).

Actualmente en el golf y pitch&putt se continúan utilizando en la mayoría de ocasiones métodos clásicos de otras épocas para registrar las partidas en torneos oficiales, como apuntar los golpes/puntos en tarjetas de cartón a lápiz, que posteriormente se han de entregar a los jueces y estos manualmente han de hacer los cálculos en función de la modalidad del torneo, handicap de cada jugador y convertir a puntos todas las tarjetas del torneo de forma manual.

Con este trabajo se pretende no solo sustituir este método clásico por una app en un móvil, sino que además hacer que sea una herramienta más a utilizar en el juego, proporcionando cierta información útil a los jugadores, no únicamente sustituir a la tarjeta y lápiz, sino dar un valor añadido a lo que actualmente utilizan.

Y en relación a otras apps que actualmente existen similares, aunque se verá con más detalle en el apartado de análisis de la competencia, el punto diferencial será el que se proporcionará información detallada de los recorridos de varios campos de Cataluña, que no están en las aplicaciones actuales, consejos de palo a utilizar.

### 1.1.1 Reglas básicas del juego y conceptos

#### **Que es el pitch & putt ?**

Es una modalidad del golf que se juega en campos de distancias más cortas, donde los hoyos nunca superan los 120m de longitud. Las reglas son las mismas que para el golf, salvo alguna pequeña diferencia como en la zona de salidas.

#### **Donde se anota actualmente la puntuación**

En tarjetas de cartón como esta:

JUGADOR	HCP P&P										COMPETICIÓN										DATA		/ / 20	
FORATS	1	2	3	4	5	6	7	8	9	ANADA	10	11	12	13	14	15	16	17	18	TORNADA	TOTALS			
RECORREGUT CATALA	96	108	70	76	80	97	73	69	104	773	90	74	56	78	81	68	66	88	82	683	1456			
RECORREGUT CATALA II	96	108	70	76	80	97	73	69	87	756	90	74	56	78	81	68	66	75	82	670	1426			
RECORREGUT EPPA	61	90	70	76	59	76	73	69	65	639	86	60	56	62	59	54	58	61	65	561	1200			
HCP	6	1	9	14	7	12	4	18	2		16	11	13	8	15	17	10	3	5					
TEMPS DE PAS MAX. de 4 JUGADORS	8	18	26	34	42	50	59	1:07	1:15		1:23	1:31	1:39	1:47	1:55	2:03	2:11	2:21	2:30		BRUT	NET		

**NORMATIVA DE JOG**

- El joc es regia per les Regles del Golf i per les normes pròpies del Centre de Golf Lloret Papalus.
- El centre no es fa responsable dels accidents de les persones alienes al joc, ni per la mala pràctica de i regori.
- Arreglau els "piquets" en els "greens", reposau les florsques en els camers, deixau en bon estat els "bunkers".
- Per respecte als demés jugadors, no atoleu la vista.
- No escà permes jugar amb boles de pràctiques en el camp.
- Si el vostre joc es lent, heu de cedir el pas.
- Vigileu molt especialment la vostra seguretat i la dels demés jugadors: no jugau mai si hi han persones davant.

EL CENTRE DE GOLF LLORET PAPALUS ES RESERVA EL DRET DE NO PERMETRE SORTIR AL CAMP SI NO ES DEMOSTRA UN MÍNIM DE PRÀCTICA I CONEIXEMENT DE LES REGLES DEL JOG I NORMES D'ETIQUETA. TAMBÉ ES RESERVA EL DRET D'EXPULSAR O PROHIBIR JUGAR EN CASOS DE MALTRATAMENT DEL CAMP O DE COMPORTAMENT INADEQUAT SEGONS CRITERI DEL PERSONAL DE CAMP.

**NORMATIVA DEL JUEGO**

- El juego se regia por la Reglas del Golf y por las normas propias del Centro del Golf Lloret Papalus.
- El centro no se hace responsable de los accidentes de las personas foráneas al juego ni por la mala práctica del juego.
- Arreglad los "piquets" en los "greens", reposad la "chuletas" en las cestas, dejad en buen estado los "bunkers".
- Por respeto a los demás jugadores, no atoleis la vista.
- No está permitido jugar con bolas de prácticas en el campo.
- Si vuestro juego es lento, tened que ceder el paso.
- Viglad muy especialmente vuestra seguridad y la de los demás jugadores, no juguéis nunca si hay personas delante.

EL CENTRO DE GOLF LLORET PAPALUS SE RESERVA EL DERECHO DE NO PERMITIR SALIR AL CAMPO SI NO SE DEMUESTRA UN MÍNIMO DE PRÁCTICA Y CONOCIMIENTO DE LAS REGLAS DEL JUEGO Y NORMAS DE ETIQUETA. TAMBÉN SE RESERVA EL DERECHO DE EXPULSAR O PROHIBIR JUGAR EN CASOS DE MALTRATAMIENTO DEL CAMPO O DE COMPORTAMIENTO INADECUADO SEGUN EL CRITERIO DEL PERSONAL DEL CAMPO.

*Introducción. 1 Tarjeta de puntuación*

**Handicap del jugador (HCP)**

Cada jugador se le asigna un HCP en función de su nivel de juego. Es un número decimal cuyo valor máximo puede ser 21. No hay jugadores con un handicap mayor que este, y no hay mínimo pudiendo haber handicaps negativos, y con decimales.

El HCP son los golpes de ventaja que se les da a cada jugador, en función de su nivel para así puedan competir con jugadores de distinto nivel. Cuanto más alto es el HCP, indica que el jugador tiene peor nivel de juego, y por tanto se le conceden más golpes de ventaja.

El HCP puede tener decimales, lo que se hace es redondear su valor para traducirlo en golpes. Un jugador con 13,4 se le tendrá en cuenta como HCP 13, y un jugador con 13,6 será equivalente a HCP 14. Para 13,5 se redondea a la baja.

El HCP puede bajar o subir a medida que se juegan torneos oficiales en función de los resultados, en partidos amistosos no se tiene en cuenta.

**Handicap del hoyo**

A cada hoyo del campo (hay 18 hoyos en un campo completo), se le asigna un handicap del 1 al 18. Este handicap indica la dificultad del hoyo, y se asigna por el propietario del campo, y ya no suele variar.

Este handicap se utiliza para calcular donde tendrá cada uno de los jugadores los golpes de ventaja en función del handicap de cada jugador.

Un ejemplo: Un jugador con handicap 12, indicará que tiene 12 golpes de ventaja antes de empezar el partido. Estos 12 golpes se repartirán entre los hoyos que tengan handicap

del 1 al 12, quiere decir que a este jugador se le restará 1 golpe de los que haya dado en cada uno de esos hoyos, en los hoyos con handicap asignado del 13 al 18 a este jugador se le contarán los golpes que de sin aplicarle ninguna ventaja.

### **Par del hoyo**

El par significa los golpes que se ha establecido por consenso como que son suficientes para embocar la bola en cada hoyo.

En el pitch && putt todos los hoyos son par 3, independientemente de su distancia.

Quiere decir que se ha establecido el acuerdo que con 3 golpes sería suficiente para embocar la bola.

En base al par del hoyo y los golpes de más o de menos que se necesite para embocar, el jugador estará *x golpes sobre par* (si ha hecho x golpes mas que el par definido), o *x golpes bajo par*, si ha hecho menos que el par.

### **Licencia**

Para poder jugar torneos oficiales al Pitch && Putt es necesario que el jugador tenga un número de licencia, que le cubre además con un seguro en caso de accidente. En el caso que el jugador no tenga licencia, porque no juega torneos oficiales, durante el registro podrá indicar cualquier cadena de texto como si fuera un nombre de usuario para poder registrarse y hacer login posteriormente en la app.

### **Distancias posibles**

En los campos catalanes & europeos hay dos posibles distancias en los recorridos del campo.

- EPPA: European Pitch & Putt Association
- ACP: Asociacio Catalana de Pitch & Putt

La primera suele tener longitudes más cortas que la segunda, y normalmente no se permite jugar con más de 3 palos.

Pero las normas de puntuación y juego aparte de eso son las mismas para ambas.

Un mismo campo puede tener ambas distancias teniendo dos zonas de salida diferentes para un mismo green.

### **Modalidades posibles** [\[ref. 1\]](#)

Hay diferentes modalidades individuales posibles variando la forma de calcular los puntos/ganador. Hay modalidades que cuentan los golpes, hoyos ganados y otras modalidades que calculan puntos en función de los golpes.

Independientemente de la modalidad escogida, en la app siempre el jugador apuntará golpes, y será la app la que calculará puntuaciones y ganadores en función de los golpes, modalidad, handicap del jugador y handicap del hoyo. Así facilitamos el uso de la app de una forma uniforme, y evitamos que el jugador tenga que estar calculando nada.

### **Stableford**

Modalidad por puntos. Gana quien suma más puntos.

1. Si embocamos la bola en un golpe más de nuestro handicap ,cogemos 1 punto .
2. Si embocamos la bola en dos golpes o más por encima de nuestro handicap, tenemos 0 puntos.
3. Si embocamos la bola en el hoyo con un golpe menos de lo establecido en nuestro handicap, ganamos 3 puntos.

### **Medal Play**

Modalidad por golpes. Gana quien suma menos golpes en todo el recorrido.

### **Match Play**

Modalidad por golpes en cada hoyo. Cada hoyo ganado es un punto. Gana quien suma más puntos por hoyos ganados.

## **Clasificaciones / Ganadores**

Normalmente siempre se hacen dos clasificaciones y puede haber dos ganadores diferentes:

- Por HCP: Se tiene en cuenta el handicap de cada jugador para calcular resultados netos
- Por Scratch: No se tiene en cuenta el handicap, y se mira el resultado bruto de cada jugador

## **1.2 Objetivos del Trabajo**

Dar a los jugadores una herramienta para :

- Registrar los golpes de cada hoyo de la partida
- Permitir guardar sus puntuaciones en una base de datos central, para poder ser consultada

- Facilitar los cálculos por modalidad y handicap del jugador automáticamente en base a los golpes, indicando en cada momento el jugador que va ganando por handicap y scratch
- Herramienta/Utilidad para los jugadores ofreciéndoles información del hoyo en juego, metros hasta el green, información del campo,...

Los objetivos del trabajo, se conseguirán por medio de las siguientes funcionalidades que ofrecerá la app:

- registro de los jugadores : nombre, licencia, handicap, foto
- nueva partida: fecha, nombre jugadores, handicaps, campo
- iniciar partida: registrar golpes por cada hoyo, información de cada hoyo, posición actual de la bola respecto al green en juego,
- resumen partida: ver la puntuación actual de los hoyos
- fin partida :envíe los resultados para ser registrados en una base de datos centralizada, indicar puntuaciones finales y ganador.
- información del hoyo en juego: metros, handicap, información del green, fotos del green y salida, metros hasta green en base a posición gps, palo aconsejable
- listado de campos mapeados disponibles : logo, teléfono, nombre, número de hoyos
- ficha informativa del campo : con información ampliada del campo nombre, teléfono, logo, localización, número de hoyos, vestuarios,...
- historial de partidas

## 1.3 Enfoque y método seguido

Para la elección del enfoque de como implementar la app he tenido en cuenta los siguientes condicionantes:

- App enfocada únicamente a dispositivos móviles, no hace falta preocuparse de que se vaya a ejecutar en tablets o en PCs, esta app es una tool para llevarla encima mientras se juega en el campo.
- La app se ha de poder utilizar off-line sin conexión a wifi y a veces incluso sin buena cobertura 3G. Ya que se utiliza en campos de golf algunos situados en poblaciones aisladas con cobertura de datos limitada.
- La app necesita utilizar funcionalidades de hardware específicas del terminal, como el posicionamiento por GPS.
- iOS vs Android: Para una primera versión esta app está enfocada únicamente a Android, ya que por lo que veo cubriría la mayoría de usuarios de pitch and putt. Y los usuarios con iPhone suelen ser dispositivos de empresa, y en muchos de ellos tampoco se les permite instalar apps de ocio en estos terminales.

Con estos condicionantes, opté finalmente por seguir la estrategia de desarrollar el producto en una app nativa en Android, que es por lo que he visto el tipo de móvil más utilizado en los usuarios de pitch and putt.

Descartada la opción de desarrollos web / web para móviles ya que siempre sería necesaria conexión wifi o de datos, y no se puede asegurar que en todos los campos se disponga. La app ha de funcionar, aunque esté en offline.

Tampoco hay que preocuparse por hacer un desarrollo web responsiva a diferentes tamaños de pantalla del dispositivo, es un producto enfocado únicamente a móviles, para ser usado en el campo de juego. Por tanto, tampoco hay que preocuparse de que haya consistencia de usabilidad entre distintos dispositivos.

## 1.4 Planificación del Trabajo

### Recursos

Para realizar este trabajo será necesario un computador, software, dispositivo físico para pruebas en real, y las pruebas de usuario serán en un campo de golf.

Desarrollo :

- Laptop ASUS 8GB RAM, 1.5GHz, 64-bits
  - Sistemas operativos : Windows 8 & Ubuntu 16
  - Software:
    - Desarrollo app : Android Studio (instalado en Ubuntu)
    - Memoria & documentación: MS Office (Word & Excel)
    - Plan: MS Project
    - Prototipado : [www.fluidui.com](http://www.fluidui.com)
    - Diseño: [www.draw.io](http://www.draw.io)
    - Wireframes: [www.lucidchart.com](http://www.lucidchart.com)

Tests (en dispositivo físico):

- Nokia 5 (Android 7.1.1)

Datos de la app y pruebas en real :

- Campos pitch and putt Sant Cebria y Golf El Valles





## **Fase de Diseño**

En esta fase se mostrará el diseño de la aplicación, no solo a nivel de UI, con wireframes y prototipado, sino también diseño conceptual, de la estrategia de como se muestra el contenido, y también una sección de diseño técnico mostrando la estructura de los datos, y arquitectura de la aplicación y conexiones.

- Escenarios de uso
- Diseño conceptual
  - Estrategia de contenido
  - Estructura de navegación
  - Wireframes
- Prototipado (fluidui.com)
- Diseño técnico
  - Definición de los casos de uso
  - Arquitectura

## **Fase de Implementación**

En esta fase se implementará la app, primero la interfaz de usuario, y posteriormente las funcionalidades e integración con los servicios API que sean requeridos. Y posteriormente una fase de tests funcionales, y una revisión de código y logs para limpiar posible código muerto o logs innecesarios.

- Implementación UI & navegación (activities)
- Implementación funcionalidades (libs)
- Tests
- Code review & Logs review

## **Fase de Entrega**

Preparación de la release, instrucciones de instalación, video presentación y documento de usuario.

- Pruebas en real / campo
- Manual de usuario
- Instrucciones compilacion/libs (si necesaria)
- Preparación release
- Video presentación

## 2 Fase de Requerimientos

En esta fase he incluido las secciones que tratan de: los usuarios objetivo, ficha de usuarios objetivo, contexto de uso, mapa de experiencia del usuario, indagación de requisitos mediante encuesta, un análisis de la competencia y el valor añadido que se pretende dar con esta solución

### 2.1 Usuarios

Para recopilar los datos de esta sección me he basado en más de 10 años de experiencia jugando al pitch & putt y más de 100 torneos jugados. Creo que clasificaría el 90% de los jugadores de esta modalidad del golf en dos categorías, un grupo de personas mayores (alrededor de la edad de jubilación), y otro grupo de personas de mediana edad (35 a casi 50 años). El rango de edad de más jóvenes es bastante reducido, supongo por el tema de coste y porque es posible que sea un “deporte/juego” demasiado tranquilo para según que rango de edad.

El juego/torneo se suele jugar en grupos de 4 personas, que los asignan juntos en función de su *hándicap* (ventaja que se le da a cada jugador en función de su nivel de juego), y durante el juego que dura casi 3h (más el tiempo de la cerveza postpartido), da tiempo a entablar conversaciones no sólo en relación al golf o el partido, y ver un poco las inquietudes personales de cada una de las personas del grupo, y motivaciones para jugar y en general en la vida. En base a estas conversaciones durante estos años, y a una encuesta realizada para este proyecto a dos jugadores, uno de cada grupo, he completado dos fichas tipo de estos dos grupos de usuarios, y así reflejar sus necesidades, motivaciones e inquietudes.

Desde un aspecto cuantitativo en cuanto al número de usuarios potencial, aunque pueda parecer un juego desconocido para la mayoría de personas (que muchas lo confunden con el “mini golf”) para dar una idea del grupo de usuarios que estarían interesados, actualmente las licencias de pitch and putt en Cataluña van ya por el número 30.000, hay 36 campos oficiales solo en Cataluña y se celebran más de 100 torneos todos los meses. Y en cuanto a la distribución de los jugadores entre los dos grupos establecidos apostarí por un 65% (jugadores seniors) y un 35% (jugadores mediana edad/jóvenes).

#### 2.1.1 Fichas de usuario

A continuación, he realizado dos fichas de estos dos grupos mayoritarios de usuarios, agrupando las características comunes de su perfil, objetivos, inquietudes con dos ejemplos tipo.

### Ficha usuario: Manuel (senior)



Nombre: Manuel  
Edad : > 65  
Trabajo: Jubilado  
Familia : Mujer e hijos mayores  
Inquietudes: Disfrutar del tiempo

#### Bio

Manuel es un jubilado que trabajó en una factoría de coches durante 40 años, tiene mujer, dos hijos y dos nietos. Tiene una cómoda situación económica que le permite ir a jugar semanalmente su partida de pitch & putt con sus amigos, y le gusta hacer un viaje al año organizado a Canarias o Baleares.

#### Motivación

Poder ver como sus hijos progresan, y sus nietos crecen sin demasiados problemas económicos

De vez en cuando comprarse algúnartilugio nuevo de golf, y ver si mejora en su juego de golf respecto a sus amigos

#### Personalidad



Trait	Score
ANALITICO	1
METICULOSO	1
INQUIETO	2
EXTROVERTIDO	4

#### Objetivos

Disfrutar de su partida semanal  
Disfrutar de su viaje al año  
Llevar una vida cómoda y tranquila

#### Hobbies

Golf / Pitch & Putt  
Nadar en el mar  
Caminar con la mujer por la montaña

Fase de Requerimientos.6 Ficha de usuario senior

### Ficha usuario: Pau (mediana edad)



Nombre: Pau  
Edad : 42  
Trabajo: Pequeño empresario  
Familia : Mujer & 1 hijo  
Inquietudes: Nuevas tecnologías

#### Bio

Pau es pequeño empresario, tiene mujer e hijo. Le gusta escaparse de sus obligaciones siempre que puede para jugar al golf, es una pequeña obsesión con la que se encuentra a gusto por su forma de ser meticuloso y analítico. Juega de 4 a 6 veces por semana.

#### Motivación

Reconocimiento social a través de ganar más dinero.

Mejorar su juego en golf (a veces obsesivamente) y poder dedicarse a ello plenamente algún día

#### Personalidad



Trait	Score
ANALITICO	5
METICULOSO	4
INQUIETO	4
EXTROVERTIDO	7

#### Objetivos

Ganar dinero y poder dedicar mas tiempo al golf.  
Móvil ultima generación cada año.  
Renovar palos de golf nuevos cada año  
Coche nuevo cada 3 años.

#### Hobbies

Golf / Pitch & Putt  
Coches deportivos

Fase de Requerimientos.7 Ficha usuario mediana edad

## 2.2 Indagación de requerimientos

Aparte de basarme en mi experiencia durante estos años, he hecho una pequeña encuesta sobre que funcionalidades apreciaría más cada uno de este par de usuarios de cada grupo, y aquí están los resultados.

**“Qué aspectos/funcionalidades valoraría de una nueva app para el juego?”**

Funcionalidad	Senior	Mediana edad	Comentarios
Permita registrar los golpes & calcule automáticamente puntuaciones	★★★★★	★★★★★	Es la funcionalidad principal de la app. Evitar el lápiz y papel y cálculos manuales. Ambos la valoran bien.
De información del campo	☆★★★★	☆☆★★★★	El usuario de mediana edad suele consultar también esta info a través del website de cada campo, por eso la aprecia menos.
De información del hoyo en curso	★★★★★	★★★★★	Ambos valoran bien que de información/consejos del hoyo en curso. A veces las tarjetas de los campos no incluyen suficiente info.
Disponible para iOS	☆☆☆☆★	☆☆★★★★	El usuario senior no valora que este para iOS, este tipo de usuario suelen utilizar teléfonos Android en su mayoría.
Permita guardar partida	★★★★★	★★★★★	Ambos valoran positivamente el que se permita guardar partidas, evitan tener que pasar a algún excel en su casa.
Permita consultar histórico	☆☆★★★★	★★★★★	El usuario de mediana edad suele ser algo más competitivo/meticuloso y le gusta consultar su progresión.
UI sencilla	★★★★★	☆☆★★★★	El usuario senior valora más que la UI sea sencilla y directa
UI incluir más opciones	☆☆☆☆★	☆☆★★★★	El usuario de mediana edad no le importaría que se incluyeran más opciones en la app, aunque suponga complicarla y tener que navegar. Está más acostumbrado a las nuevas tecnologías.
Informe metros hasta el hoyo	☆☆★★★★	☆☆★★★★	Ambos valoran bien esta funcionalidad para el cálculo de palo a escoger
Funcione off-line	★★★★★	★★★★★	Ambos valoran con la máxima puntuación el que funcione off-line, ya saben que a veces se pierde la cobertura de datos en según qué campos.

Fase de Requerimientos. 8 Encuesta funcionalidades

## 2.3 Contexto de uso y Customer Journey Map

La aplicación se plantea como una tool para utilizar en su mayor parte de funcionalidades durante la celebración de un partido, en el campo de pitch & putt, por ejemplo, el registro de golpes, y consulta de info de cada hoyo. Aunque también ofrecerá unas funcionalidades que es posible que se utilicen antes del partido y antes de llegar al campo de juego, como por ejemplo ver la información del campo y el recorrido de hoyos, y algunas otras funcionalidades que es posible que el usuario utilice después del partido y en su casa como por ejemplo la consulta de histórico de partidas.

En cada uno de estos contextos de uso, pueden darse unos condicionantes técnicos y de usabilidad a tener en cuenta tanto en el diseño de la app como en la arquitectura a definir de la aplicación. Se podrían agrupar en dos grandes contextos de uso de la aplicación:

Contexto	Funcionalidades	Aspecto	Condicionantes
Uso durante el juego (en el campo)	Inicio partida Resumen partida Hoyo en juego Registro puntuación Fin partida	Diseño	Se plantea esta app/producto para ser utilizada en su mayoría de funciones en el campo, durante el juego, por tanto, <b><u>su uso se restringirá al smartphone, no es necesario tener en cuenta diseño adaptativo para tabletas o PCs</u></b>  El jugador lleva el móvil, con la app abierta, pero también lleva la bolsa con los palos, y tiene que estar atento al juego del resto de jugadores, a su bola y siguiente golpe. Hemos de <b><u>facilitar la usabilidad</u></b> al jugador, ha de tener una UI simple y directa sin demasiada info ni opciones innecesarias y con la info que necesita del hoyo en curso se muestre sin tener que navegar a través de la UI.
		Arquitectura	Los campos normalmente se encuentran en zonas de montaña alejadas de poblaciones grandes, con a veces muy limitada cobertura. Hay que prever que durante el juego en el recorrido del campo puede que no disponga de cobertura de datos. <b><u>La app ha de seguir funcionando aún en modo off-line.</u></b>  Se ofrece una funcionalidad relativa al hardware del móvil, como es el posicionamiento GPS respecto al hoyo en curso, por tanto, se opta por el <b><u>desarrollo de una app nativa</u></b> en lugar de desarrollo web (que necesitaría conectividad, además).
Uso antes o después del juego (fuera del campo)	Registrar usuario Consulta histórico Consulta info campo	Diseño	El usuario normalmente ejecutará estas opciones cuando aún no esté jugando, puede estar en su casa o previo al partido en el bar del campo, puede recrearse en estas opciones porque <b><u>no tiene la presión de estar jugando, puede estar concentrado en la app y por tanto podemos plantear UI más rica en detalles si fuera necesario</u></b>

		Arquitectura	Estas <b>operativas normalmente las hará fuera del campo en algún sitio con wifi o cobertura de datos</b> , hay que tenerlo en cuenta por si se quiere que estas opciones descarguen cantidades de datos, aunque se plantea de un inicio que la app pueda funcionar off-line.
--	--	--------------	---

Fase de Requerimientos.9 Contexto de uso, funcionalidades y condicionantes

En la siguiente figura he reflejado en un customer journey map, como se sentirán los jugadores con el uso de las diferentes funcionalidades, en base a lo que les aportará de nuevo respecto a lo que en estos momentos se está utilizando, en cada uno de los momentos, antes del juego, durante el juego y en el postpartido.

## Customer Journey Map



Fase de Requerimientos.10 Mapa de experiencia de usuario

## 2.4 Escenarios de uso

Teniendo en cuenta los contextos de uso descritos en el apartado 2.3, podemos contemplar los posibles escenarios de uso concretos para las diferentes funcionalidades que ofrecerá la app.

En este apartado se definen los diferentes escenarios de uso de este producto que se pueden presentar, teniendo en cuenta donde se utilizará la app en cada uno de los escenarios, y las posibles pre-condiciones que se pueden dar en cada uno de los escenarios.

Como se ha indicado que la aplicación ha de funcionar en modo off-line, se tendrá que buscar una opción de arquitectura que combine un local storage con un remote storage. En el apartado de arquitectura se darán más detalles de la opción finalmente escogida, y en este apartado para explicar los escenarios de uso nos referiremos a local (almacenamiento local en el móvil) o remote (almacenamiento

remoto centralizado, en el que será necesaria conectividad), y entre ellas habrá que establecer un mecanismo de sincronización.

**ACTORES:** Jugadores en campo

**ESTADOS/PRECONDICIONES** (en azul)

- INT: Conexion a internet
- REG: Usuario registrado
- LOG: Usuario logeado
- CAMPO: Datos de campo descargados en local

**FLUJO:** Indicado en las columnas (Jugar Partida, Grabar Partida, Histórico) (en amarillo)

**POSTCONDICIONES:** Indicado en las celdas de la tabla (en verde)

Escenario	ESTADOS (PRECONDICIONES)				ACCIONES (FLUJO & POSTCONDICIONES)		
	INT	REG	LOG	CAMPO	Jugar partida	Grabar partida	Historico
1	SI	SI	SI	SI	Firestore automáticamente sincroniza con los datos de local y se descargará si hay actualizaciones. Jugará con la última actualización de datos del campo	Partida grabada en Firestore	Historico disponible de Firestore
2	SI	SI	SI	NO	Al elegir el campo se descargarán los datos del campo. Jugará con la última actualización de datos del campo	Partida grabada en Firestore	Historico disponible de Firestore
3	SI	SI	NO	SI	La app le preguntará si quiere logearse antes para tener sus datos personales. Jugará con la ultima actualización de datos del campo	Antes de grabar le pedirá que haga login. Hasta que no haga login la partida se guardará en local	Historico disponible en Local hasta que no haga login.
4	SI	SI	NO	NO	La app le preguntará si quiere logearse antes para tener sus datos personales. Se descargarán los datos del campo (esté logeado o no). Jugará con la última actualización de datos del campo	Antes de grabar le pedirá que haga login. Hasta que no haga login la partida se guardará en local	Historico disponible en Local hasta que no haga login.
5	NO	SI/NO	NO	SI	La app detectara que no hay internet. No pedira hacer login. Jugará con los datos del campo que tenga en local	Partida se grabara en local.	Historico disponible en Local
6	NO	SI/NO	NO	NO	La app le indicará que no tiene los datos del campo en local, y no podrá jugar.		

					Le recomendará se los baje en alguna wifi		

## 2.5 Análisis de la competencia y valor añadido

Para el análisis de la competencia me he descargado varias apps de Google Play, las he estado probando y al final me he quedado con las dos que me han parecido que reúnen parte de las funcionalidades que apreciarían los usuarios, una de ellas es totalmente gratuita con anuncios, y la otra también es gratuita en su versión básica con anuncios, pero después tiene servicios “premium” bajo pago de una suscripción anual.

**ScoreCard:** App muy simple pero funcional. Gratuita

PROS	CONS
Gratuita	Tiene errores de datos, en algunos campos la numeración y longitud de los hoyos no es la correcta
Permite registrar partidas con hasta 4 jugadores y grabarlas	No marca la posición GPS ni longitud hasta el green (no se si es un bug)
Permite compartir resultado con un screenshot de la tarjeta	Solo permite registrar 9 golpes por hoyo (no debería estar limitado)
Tiene datos de algunos campos de Cataluña con los recorridos	No calcula puntuaciones en base a la modalidad del juego
Parece que permite ver en un mapa tu posición via GPS en el campo	No envía resultados a ningún storage centralizado, se guarda en local
Interfaz muy simple, buena usabilidad	No tiene información de los campos (vestuario, telefono, practicas, localización, bar,...)
Guarda historial de partidas	No tiene toda la info de los hoyos: handicap, medida EPPA, par del hoyo,...
	No permite registrar datos del usuario

*Fase de Requerimientos. 11 ScoreCard: Pros & Cons*

**Hole 19:** Está en el ranking entre las mejores apps de golf. Servicios premium bajo suscripción anual de 50 EUR.

PROS	CONS
Tiene datos de muchos campos a nivel internacional, y también los de Cataluña	Muchas de estas funcionalidades solo están bajo la versión Premium, bajo suscripción anual de 50 EUR (cálculo de puntos según handicap, medidas de distancia, mapas alta resolución, añadir notas en cada hoyo, estadísticas,...)
Detecta tu localización y te indica los campos de golf más cercanos	No tiene mucha información adicional por hoyo ni de los campos (como vestuarios, campos de prácticas, etc...)
Interfaz de usuario muy completa y rica que la anterior app.	
Permite crear tu perfil de jugador	
Incluye fotos de los campos	
Incluye widget indicando meteo de la zona del campo elegido	
Croquis detallado de cada hoyo (metros, par, hdc)	
Navegación de un hoyo al siguiente sencilla	
Marca los metros al green desde tu posición	
Conecta a Google Fit para guardar tu recorrido	
Tiene opción de sincronización para bajar los datos del campo a local y poder jugar más rápido	
Permite indicar puntos intermedios de tiro y te indica los metros desde tu posición al punto escogido	
Clasifica los golpes depende de donde has golpeado: calle, putt, bunker, penalidades.	
Muestra un resumen de tarjeta muy completo y detallado	

*Fase de Requerimientos. 12 Hole 19: Pros & Cons*

## 2.6 Funcionalidades a incluir en la app

Teniendo en cuenta el tiempo que se dispone para la fase de implementación, los aspectos y funcionalidades que más valoran los usuarios objetivo basándome en la encuesta, y algunas de las funcionalidades que he visto en las dos apps analizadas que me han parecido interesantes, la lista de funcionalidades que incluirá esta app serán las siguientes (haciendo click en el nombre de cada una navegará a su wireframe):

- **Pantalla inicial**: Mostrará 3 image buttons grandes que ocuparán toda la pantalla con las opciones/funcionalidades más usadas como “Nueva partida”, “Listado de campos”, “Historial”, para facilitar la usabilidad al jugador.

*Navegación:*

- Botón Nueva Partida: Navega a la pantalla de “Nueva Partida”.
- Botón Listado de campos: Navega a la pantalla de “Listado de Campos”
- Botón Historial: Navega a la pantalla “Historial”
- Hamburger en ActionBar: Desplegará un Navigation Drawer con todas las opciones

- **Menú principal**: Será un menú desplegable tipo navigation drawer que mostrará las opciones principales de primer nivel de la app, y nombre y foto del jugador.

*Navegación:*

- Opción Login
- Opción Registro Jugador
- Opción Nueva Partida
- Opción Listado de campos
- Opción Historial

- **Registro jugador**: Permitirá registrar al usuario en una base de datos centralizada, guardando nombre, su foto y licencia (si no tiene licencia puede informar cualquier cadena de texto a modo de usuario), y un PIN de 4 dígitos que hará de password. El estar registrado permitirá al usuario que pueda guardar sus partidas de forma centralizada. Si no se registra y logea sus partidas quedarán guardadas solo en local.

*Campos editables:*

- Foto
- Nombre
- Licencia
- Handicap
- PIN

*Navegación:*

- Botón Guardar : Guarda el jugador y vuelve a la pantalla principal.
- Flecha Back ActionBar: Vuelve a la pantalla inicial.

- **Login** : Una vez registrado, le permite entrar a su cuenta, y poder grabar sus partidas en la base de datos centralizada. Si se autentica correctamente aparecerán sus datos en la cabecera del navigation drawer.

*Campos editables:*

- Licencia
- PIN

*Navegación:*

- Botón Login : Autentica al jugador y vuelve a la pantalla principal. O sale un mensaje de autenticación fallida y vuelve a la pantalla principal.
- Botón Registrarme: Para el usuario que aún no se haya registrado, navegará hacia la pantalla de Registro de jugador.
- Flecha Back ActionBar: Vuelve a la pantalla inicial.
- **Nueva partida:** Se inicia una nueva partida, permitiendo elegir un campo con un desplegable o lista scrollable de campos, una fecha de la partida con un calendario (aparecerá pre-seleccionada la fecha actual), una distancia del campo (aparecerá pre-seleccionada la ACPP), modalidad del juego, el tee/hoyo desde el que comenzarán a jugar (aparecerá pre-seleccionado el 1), y un botón de elección de jugador oponente, que abrirá una pantalla para registrar al jugador rival, y un botón de iniciar juego, que iniciará la partida abriendo la pantalla del hoyo inicial indicado.

*Campos editables:*

- Campo (seleccionable)
- Fecha (abre calendar)
- Distancia del campo (EPPA/ACPP), si el campo tiene dos distancias
- Modalidad del juego
- Hoyo/Tee inicial

*Navegación:*

- Jugador oponente: Abrirá pantalla Registro jugador oponente
  - Botón iniciar juego: Navegará a Hoyo en Juego
- 
- **Jugador oponente:** Pantalla similar a Registro Jugador, pero no registrará al jugador oponente, simplemente grabará sus datos en la partida jugada. Se pueden recuperar los datos automáticamente de un jugador registrado informando Licencia y dando al botón autenticar. Pero no es obligatorio, se pueden informar manualmente y sin autenticar se pueden guardar en la partida.

*Campos editables:*

- Foto
- Nombre
- Handicap
- Licencia
- PIN

*Navegación:*

- Botón Guardar : Guarda el jugador y vuelve a la pantalla anterior.
- Botón Autenticar: Recupera los datos del jugador oponente registrado en funcion de Licencia y PIN informados.
- Flecha Back ActionBar: Vuelve a la pantalla anterior.

- **Hoyo en juego**: Pantalla que mostrará el hoyo que se está jugando con una serie de información del hoyo y distancia a green desde la posición actual

*Campos editables:*

- Golpes jugador y rival (mediante botones)

*Campos informativos:*

- Metros, Par, Hcp del hoyo
- Fotos: Croquis & Green
- Distancia: Muestra distancia a bandera desde posición actual.

*Navegación:*

- Botón next / prev : Navega a la pantalla del hoyo siguiente o previo.
- Botón Resumen: Navega a la pantalla Resumen Partida con los golpes actuales y puntuación.

- **Fin partida**: Mostrará ganadores

*Campos informativos:*

- Golpes y puntuación según HCP
- Golpes y puntuación según Scratch

*Navegación:*

- Botón Guardar Partida: Grabará la partida y volverá a la pantalla inicial.

- **Historial**: Permitirá consultar partidas guardadas filtrando por campos, modalidad, distancia y fecha

*Campos editables:*

- Fecha desde
- Seleccionable nombres de campos
- Seleccionable distancia
- Seleccionable modalidad

*Navegación:*

- Botón Mostrar: Navegará a la pantalla Resultados Historial enviando los parámetros seleccionados. Para que muestre las partidas que se ajusten a los valores.

- **Resultados Historial**: Listado de partidas que coinciden con los valores de filtro indicados en Historial.

*Campos informativos:*

- Fecha, campo, puntuación hcp, puntuación scratch

*Navegación:*

- Haciendo tap en un elemento de la lista llevará a Partida Historial

- **Partida Historial**: Muestra golpes, puntuaciones, datos jugador oponente, ganadores por hcp y scratch y golpes por hoyo de cada uno

*Campos informativos:*

- Nombre, Licencia y handicap jugador oponente
- Ganadores partida
- Golpes por hoyo

*Navegación:*

- Botón Historial: Vuelve a Historial.
- Botón Pantalla Inicial: Vuelve a Pantalla Inicial

- **Listado de campos**: Muestra fichas de campos scrollable, con logo, nombre, teléfono y población.

*Campos informativos:*

- Logo, nombre, teléfono, población del campo.

*Navegación:*

- Tap en un elemento de la lista navega al Ficha Campo
- Flecha Back ActionBar: Vuelve a Pantalla Inicial

- **Ficha campo**: Muestra la información del campo y localización

*Campos informativos:*

- Logo, nombre, teléfono, población del campo.
- Email & URL de la web del campo

*Navegación:*

- Botón Preview Recorrido: Navega a Preview Recorrido
- Flecha Back ActionBar: Vuelve a Listado de Campos

- **Preview recorrido**: Muestra listado de hoyos, indicando metros, par, handicap

*Campos informativos:*

- Número de hoyo, metros, par, handicap

*Navegación:*

- Tap en elemento de la lista lleva al Detalle Hoyo

- **Detalle hoyo** : Mostrará información específica del hoyo seleccionado.

*Campos informativos:*

- Número de hoyo, metros, par, handicap

- Croquis hoyo y foto del green
- Detalles en modo texto, como número de lagos, bunkers, fuera límites.

*Navegación:*

- Botón Next y Prev : Navegará al siguiente y anterior hoyo
- Botón Volver Listado: Navegará al Listado de campos

## 3 Fase de Diseño

### 3.1 Estrategia de contenido

El foco de esta app no es de presentar mucha contenido de información que se vaya publicando y variando a menudo, y la poca información que hay que presentar, al final se ha priorizado el presentarla lo más simple y accesible para el usuario sobre todo en las funcionalidades que se utilizan durante el juego.

Por otro lado, la única información que el usuario ha de entrar en la aplicación serán los golpes en cada hoyo, y para facilitar la entrada se hará mediante botones que incrementaran o decrementaran el contador, para que pueda hacerlo con el pulgar directamente.

**Información del campo:** Se presenta en una única pantalla concisa sobre los servicios que ofrece el campo.

**Información del hoyo:** Se presenta la información del hoyo en juego en una única pantalla para que el usuario no tenga que navegar para ver información adicional mientras está jugando.

**Notificaciones:** Se le presentará al usuario una notificación cada vez que haya un nuevo campo registrado en la base de datos/storage, o cuando esté disponible una nueva versión de la app.

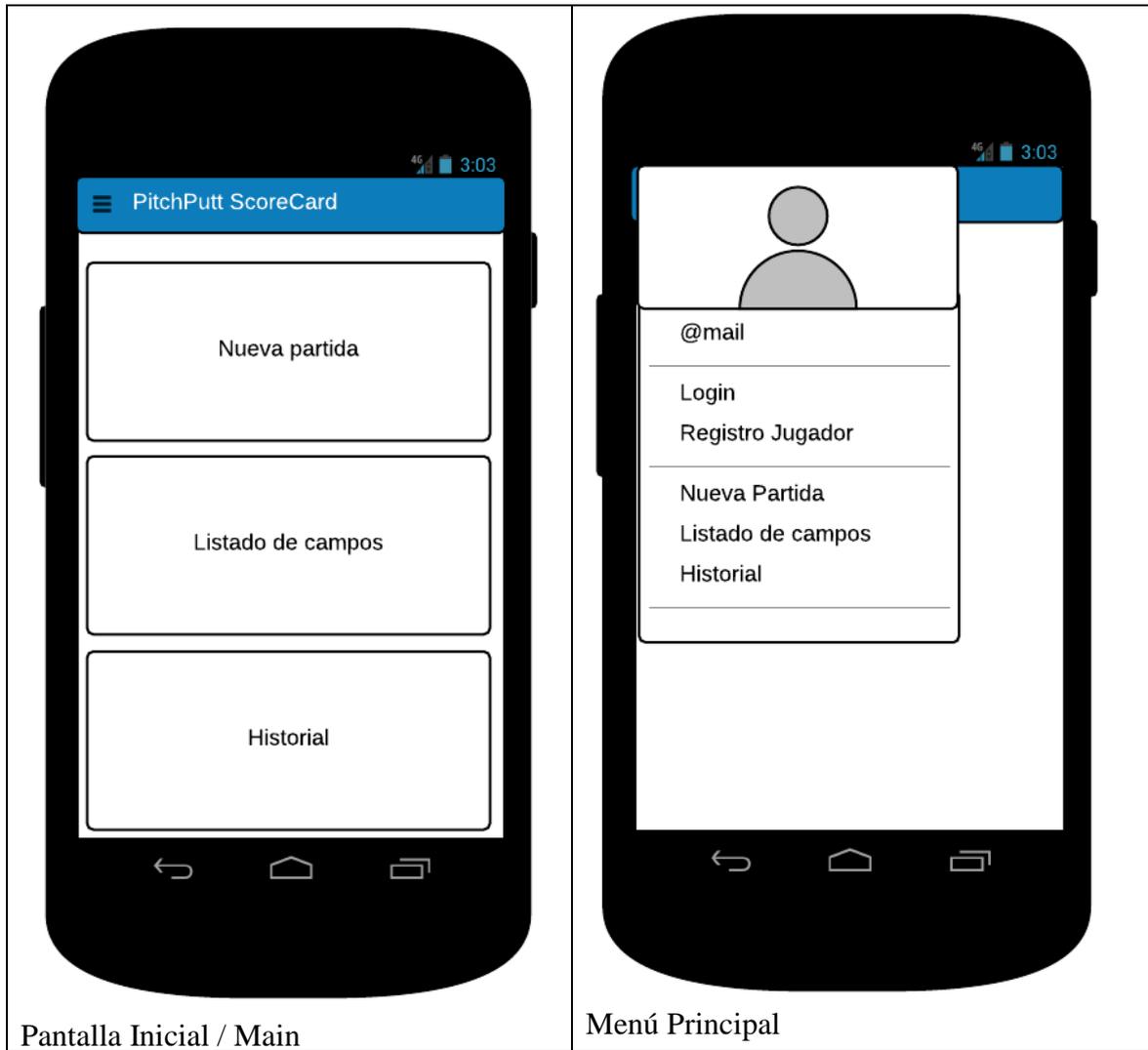
#### 3.1.1 Multiidioma

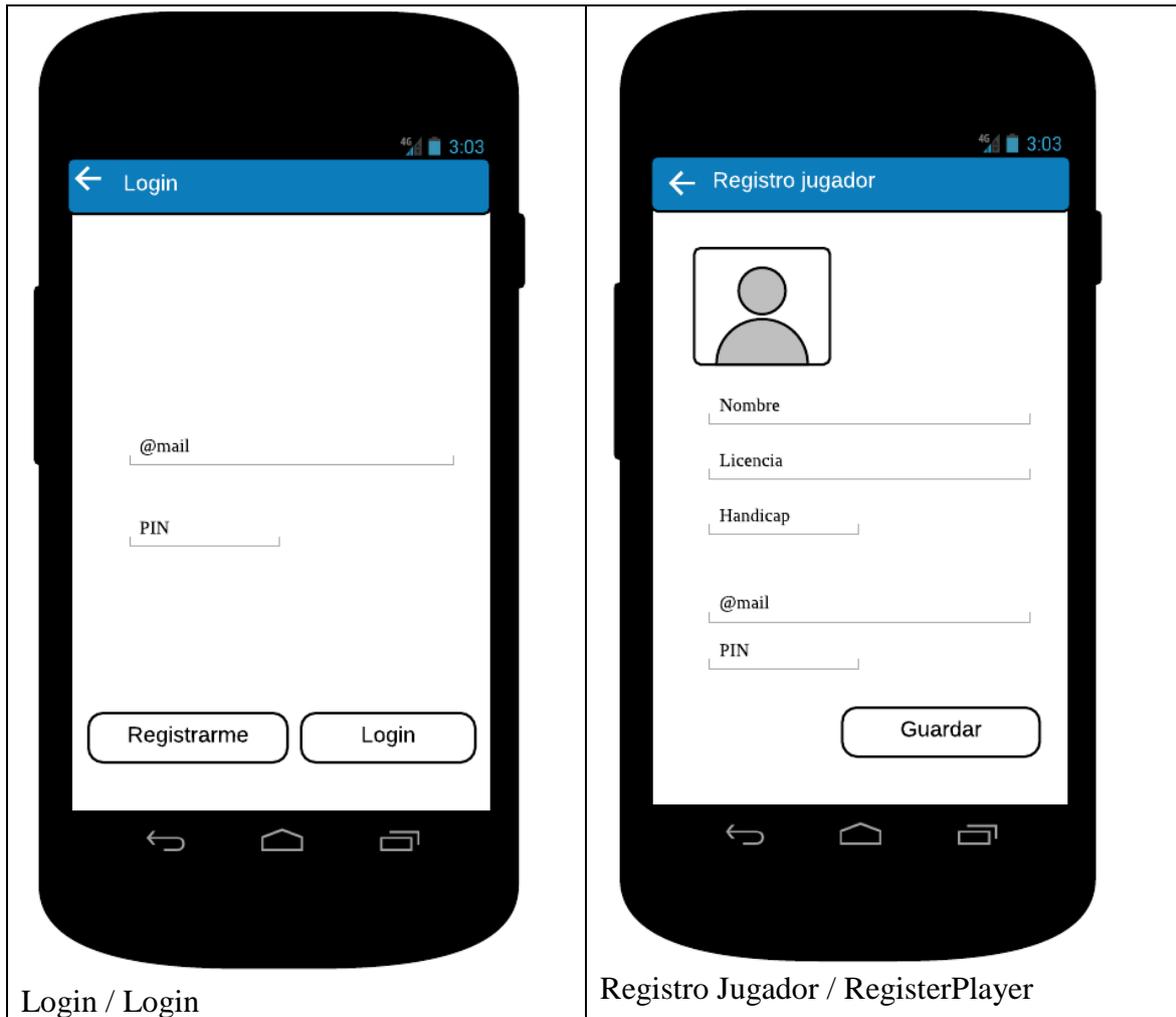
Este juego de pitch & putt es originario de Irlanda, junto con Cataluña son las dos potencias europeas del juego, y poco a poco se ha ido extendiendo por España, por este motivo la app se va a preparar en tres idiomas: Catalán, Inglés y Castellano.

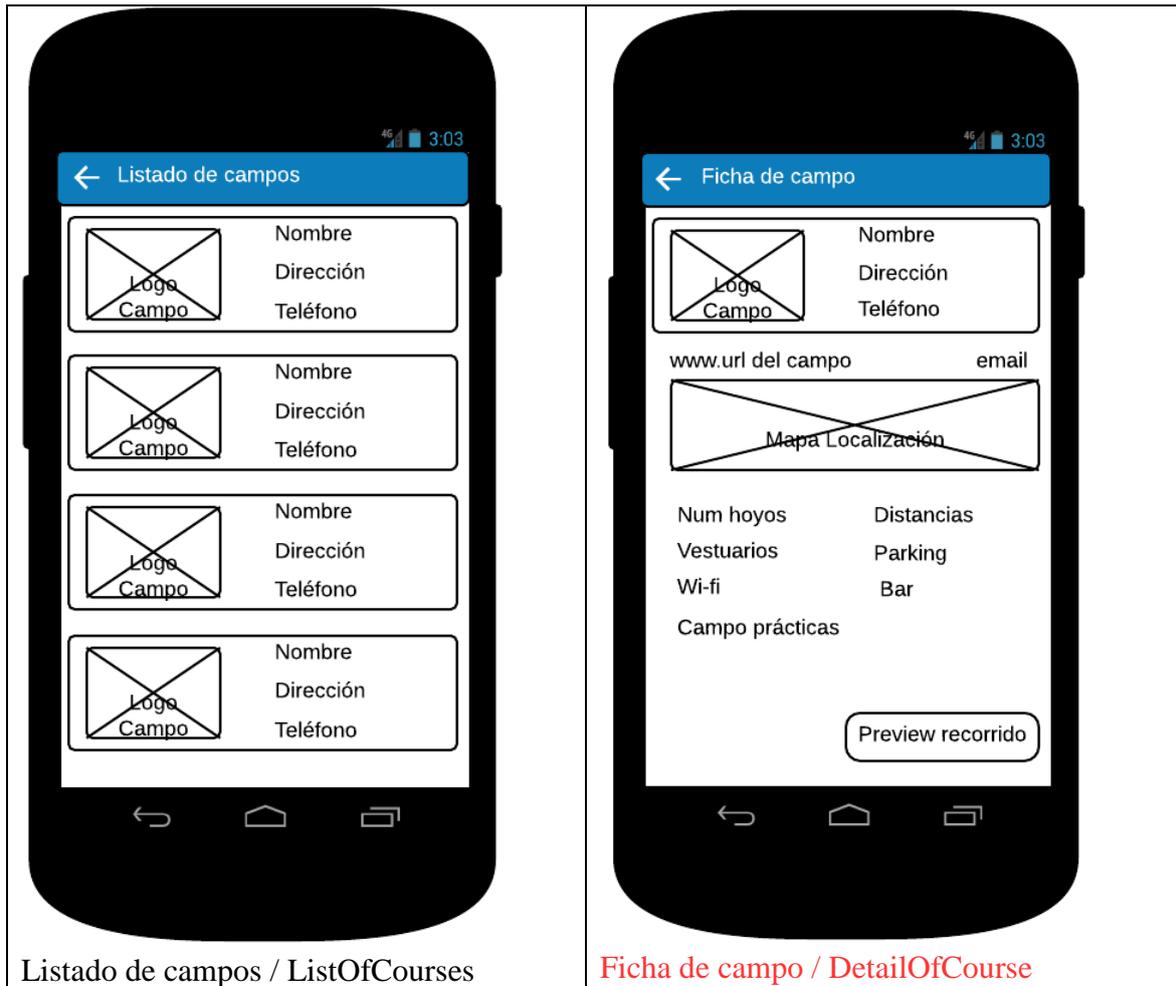
La app detectará el idioma del dispositivo, y si no es ninguno de estos tres, se mostrará por defecto en Inglés.

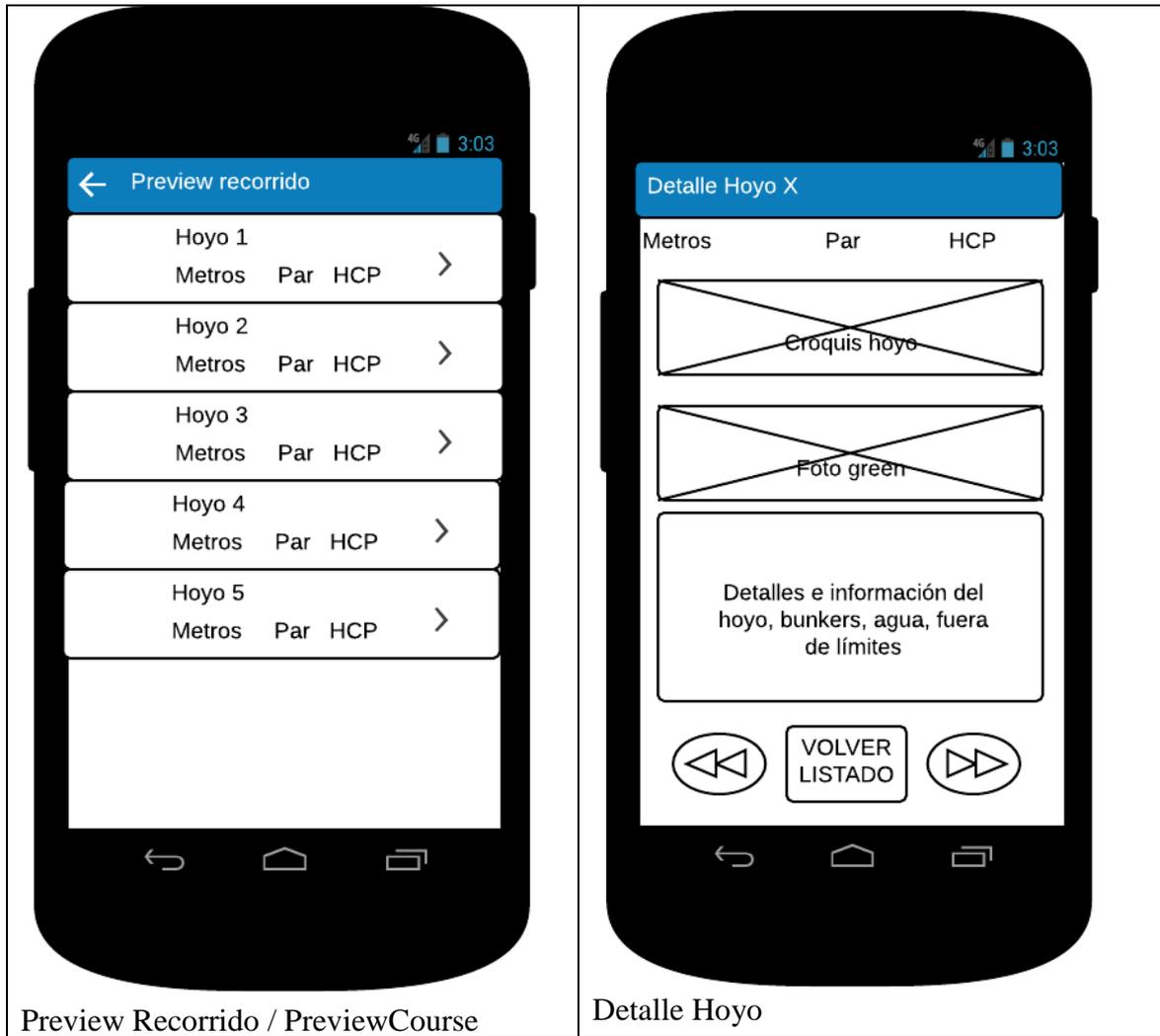
### 3.2 Wireframes

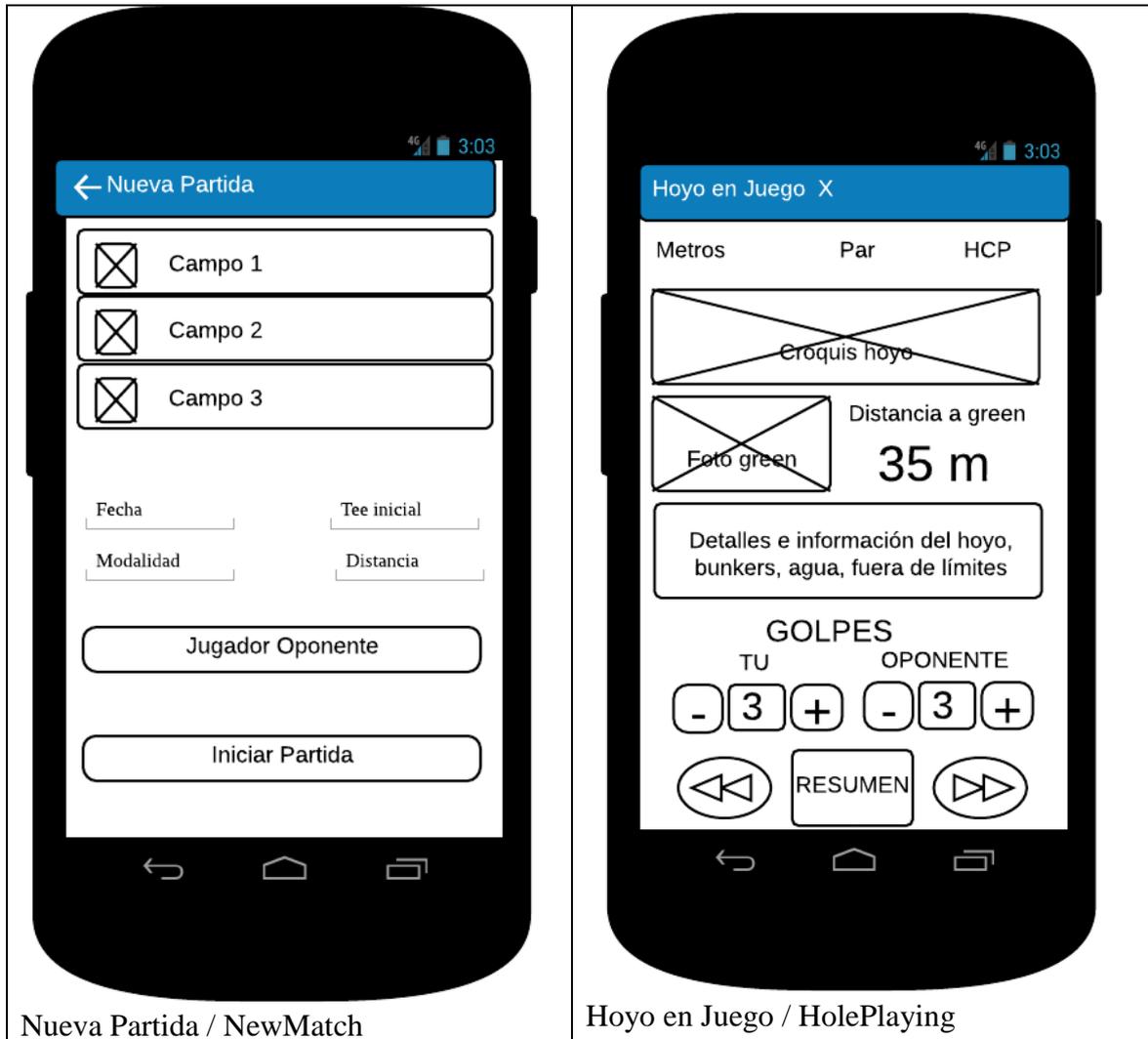
Para realizar los wireframes los primeros esbozos se hicieron a mano en papel, y cuando ya tenía un diseño definido utilicé la web [www.lucidchart.com](http://www.lucidchart.com) .

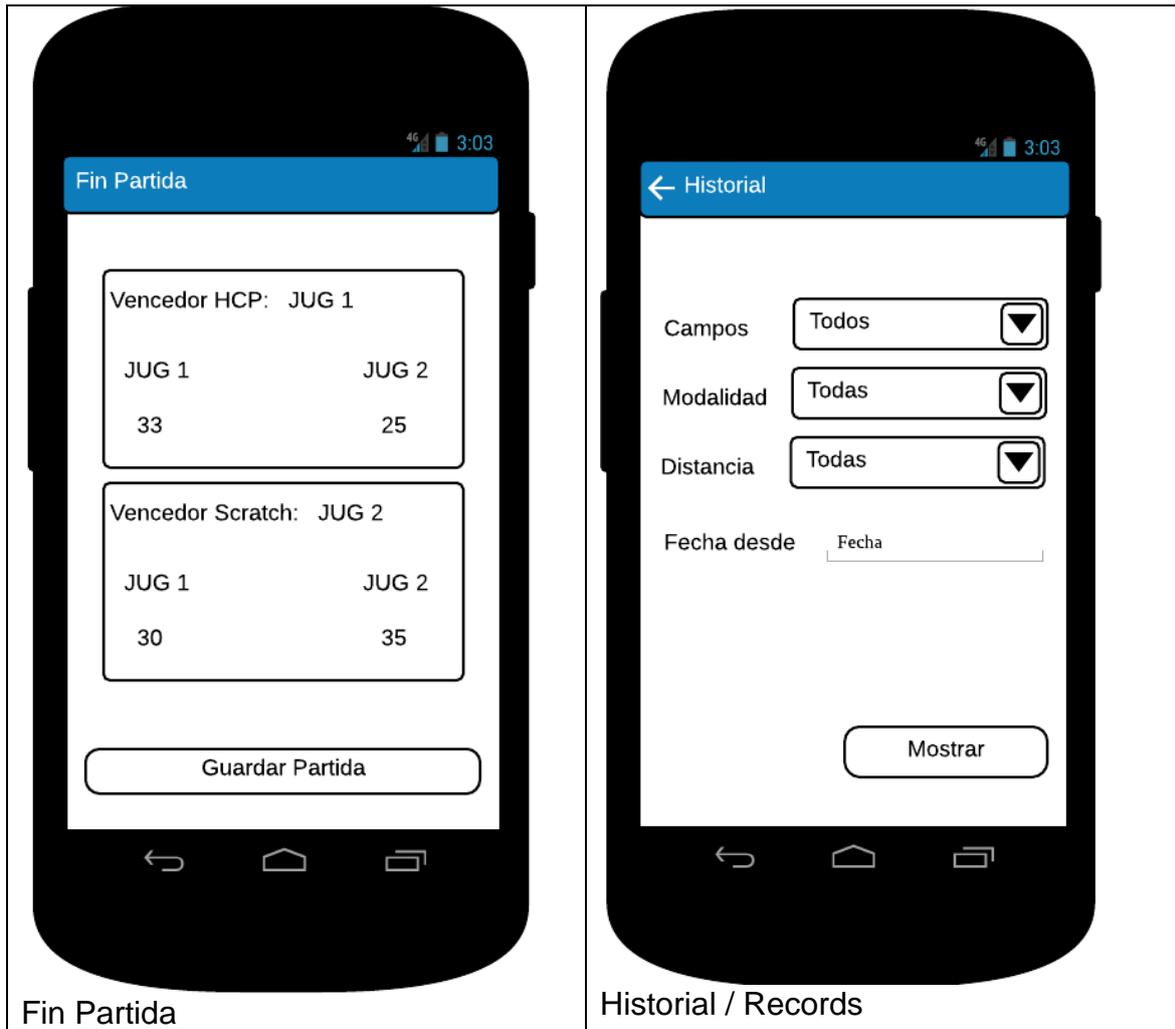


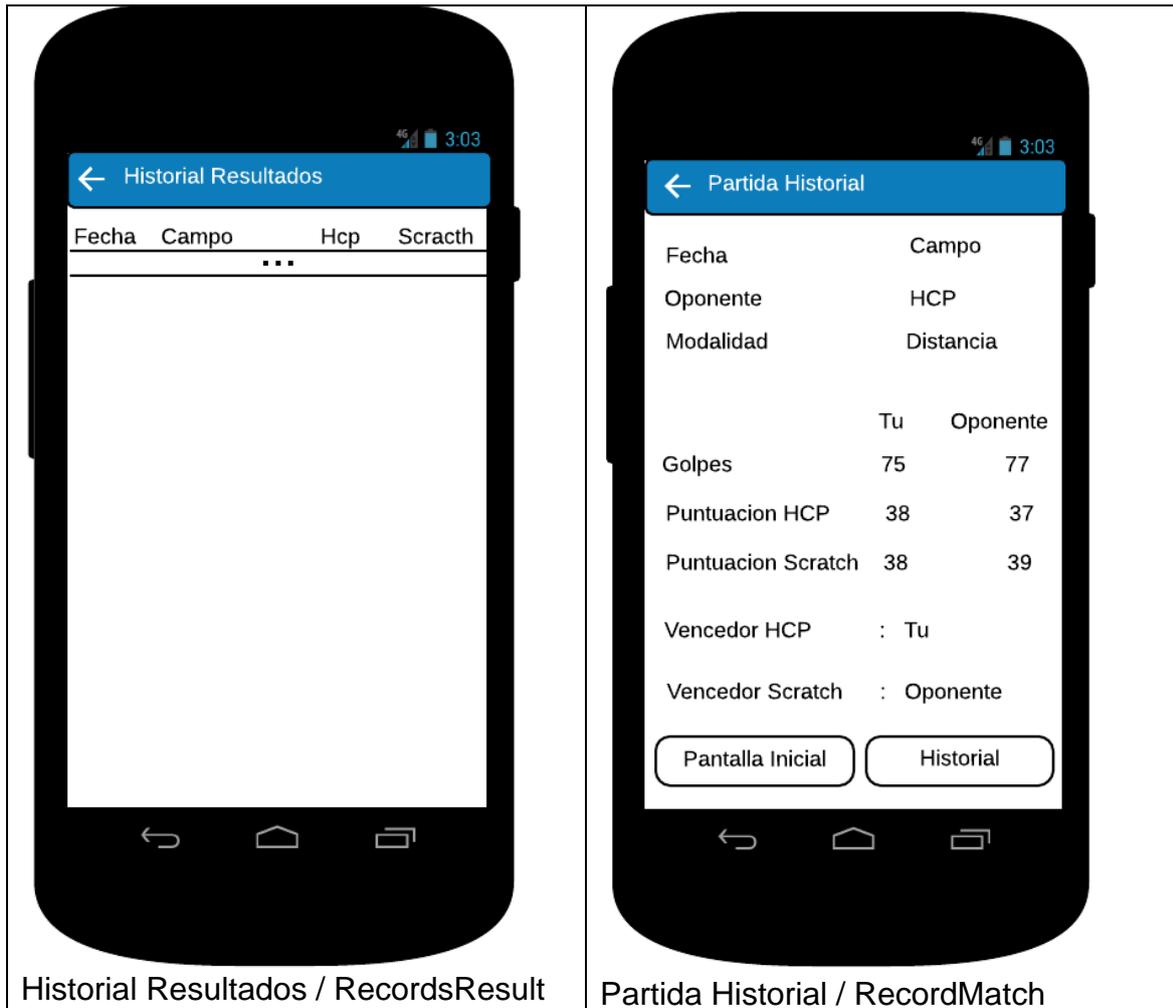


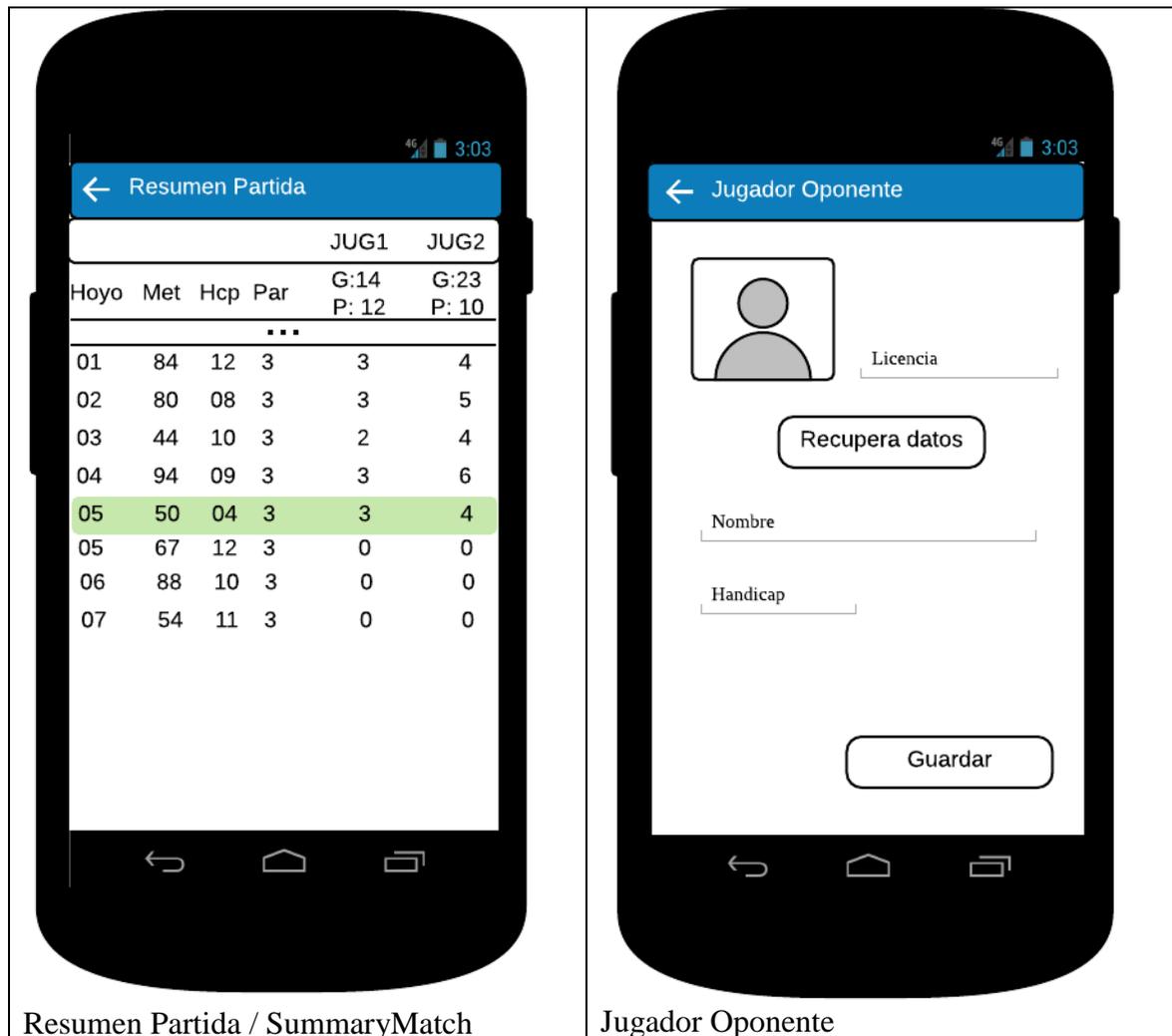












### 3.3 Navegación

El flujo de navegación se ha diseñado de forma sencilla y directa para las operativas, pantallas que se usan durante el juego, y algo más compleja para las operativas que se suelen usar previas justo al partido o fuera totalmente de juego, como consulta de historial, consulta de campos, preview de recorridos, etc...

Las operativas durante el juego realmente se pueden resumir en 2 pantallas: Hoyo en juego y Resumen Partida, la tercera pantalla utilizada Final partido simplemente muestra el resultado final del juego una vez acabado.

Elementos de navegación:

- Navigation Drawer menú: contiene las opciones del menú principal
- Pantalla Inicial: Contiene enlaces a las 3 operativas principales de primer nivel
- BackArrows en el ActionBar: Para volver a los niveles superiores
- Botoneras : Siguiete y Previo para navegar por los recorridos p.ej.

- Home: Habrá iconos en el actionBar para volver a la pantalla inicial.

**Por niveles de profundidad:** Si lo miramos por niveles de profundidad tenemos en los 3 primeros niveles las operativas más usadas, siendo el primer nivel simplemente una pantalla inicial de enlaces y el menú desplegable

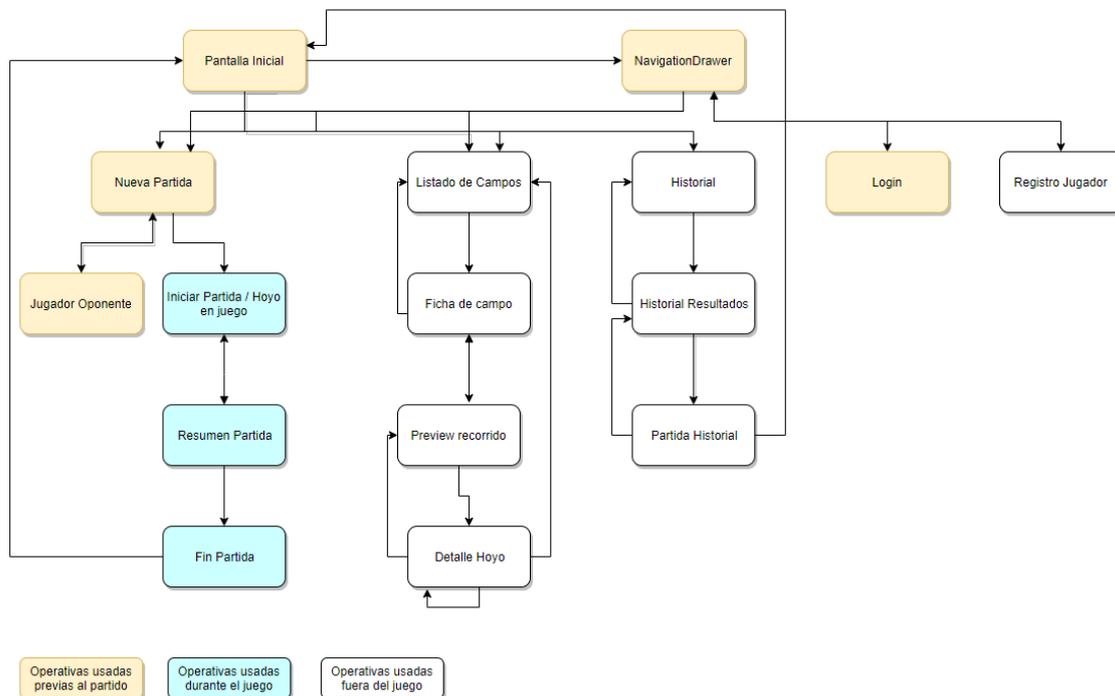
- 1º nivel: Pantalla principal, NavigationDrawer menú.
- 2º nivel: Nueva partida, Listado de campos, Historial, Login, Registro Jugador
- 3º nivel: Jugador oponente, Iniciar Partida, Ficha de campo, Historial resultados

En el resto de niveles se presentan opciones poco utilizadas, información que se consulta fuera del juego o la pantalla de resultado de fin de partida.

### Por contexto de uso:

En el diagrama de navegación se ha distinguido las operativas por el contexto en el que se utilizan: las que se utilizan justo previas al partido, antes de empezar el juego (en amarillo), las operativas que se utilizan durante el partido (en azul), y el resto que se utilizan normalmente cuando no se está jugando (en blanco).

### Diagrama de navegación:



Fase de Diseño. 13 Diagrama de navegación

## 3.4 Prototipo

Prototipo disponible a través de este link en fluidui.com

[Prototipo PitchPutt ScoreCard](#)

## 3.5 Diseño técnico

### 3.5.1 Diagrama de casos de uso

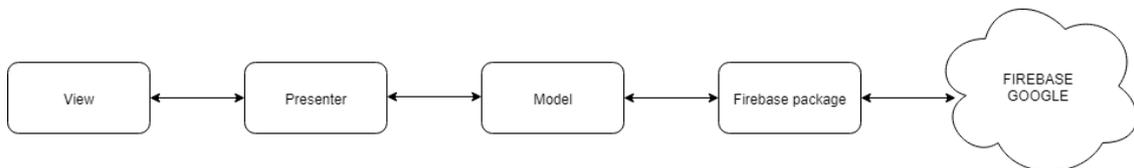


Fase de Diseño. 14 Diagrama de casos de uso

### 3.5.2 Arquitectura interna de la aplicación

El modelo seguido para diseñar la arquitectura interna de la app android estará basado en el paradigma MVP (Model-View-Presenter), donde la View se encarga de recoger la interacción con el usuario y presentar los datos, el Presenter será el componente que comunica/enlaza la View con el Model haciendo las peticiones al model y devolviendo los datos a la vista, y el Model se encargará de contener la lógica de la aplicación y la interacción con el storage mediante a la llamada a otra capa de conexión que creare que manejará el API del storage en Firebase.

Esta última capa del package de manejo del API de Firebase lo haré para aislar lo que es la lógica de la app (que estará en el modelo) de las conexiones al storage que en este caso es firebase, pero que si en un futuro cambia el servicio y es otro tipo de storage como una base de datos en un servidor, solo habrá que cambiar este último package, y el resto de los tres niveles del paradigma no habrá que modificarlos.



Fase de Diseño. 15 Tiers arquitectura aplicación

#### 3.5.2.1 Implementación física

Para implementar este paradigma seguiré este criterio para cada activity:

Crearé una interface por cada funcionalidad que contendrá las interfaces de cada uno de los niveles del paradigma MVP:

- <funcionalidad>Model,
- <funcionalidad>View,
- <funcionalidad>Presenter

Y se creará una clase java que implementará cada una de estas 3 interfaces:

- **Models:** Clase java <Funcionalidad>Model.java
- **Views:** Clase que extiende de Activity e implementará, y tendrán esta nomenclatura <Funcionalidad>View.java
- **Presenters:** Clase java <Funcionalidad>Presenter.java

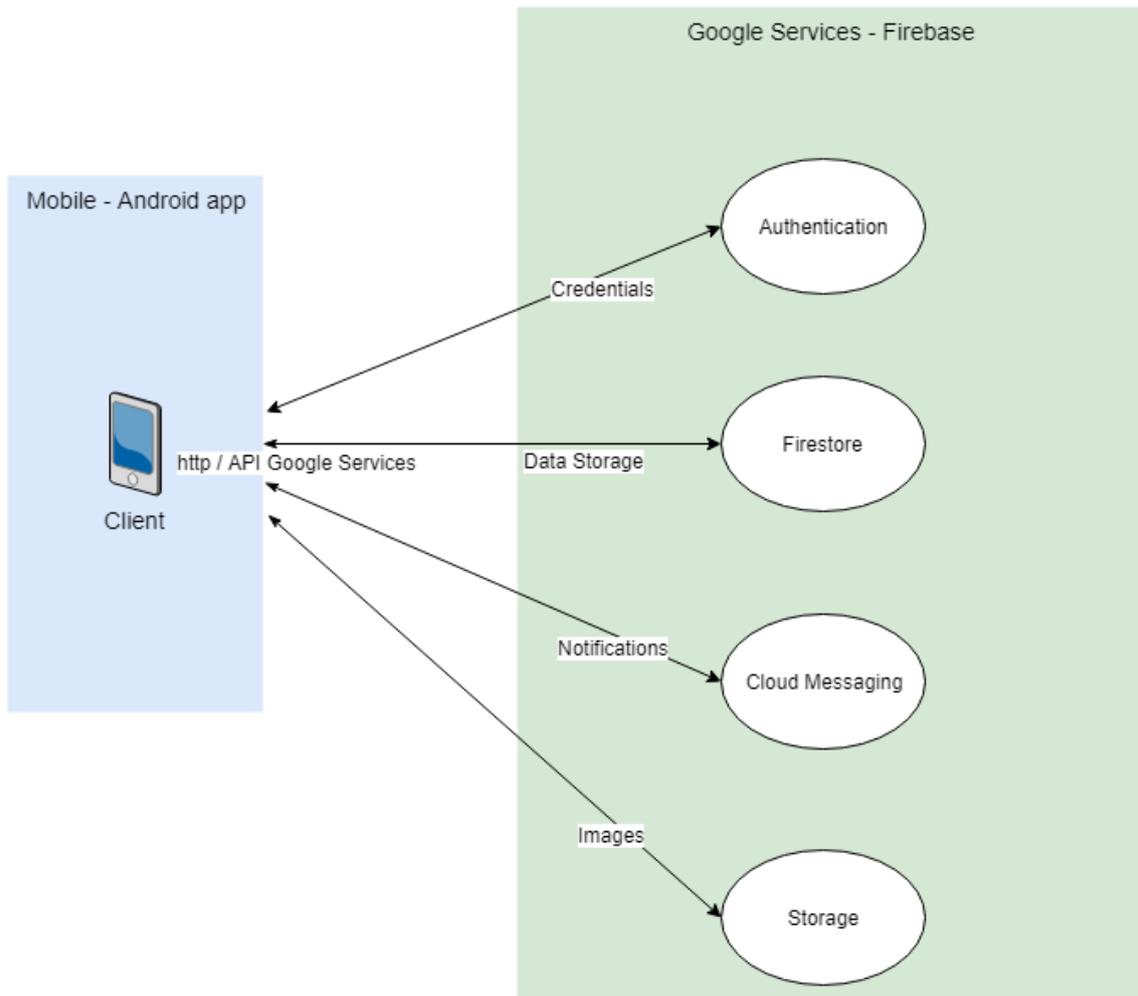
Como habrá 16 activities, y se crearán 3 clases para cada una de ellas, lo organizaré en diferentes packages, para tenerlo más organizado, así habrá los siguientes packages:

- /interfaces → contendra las interfaces
- /models → contendra las clases java modelos
- /views → contendra las clases java vistas (activities)
- /presenters → contendra las clases java presenters
- /entities → contendra las clases java entities (entidades)
- /firebase → contendra las clases java de conexion con API Firebase

### 3.5.3 Arquitectura de conexiones & servicios

La app tendrá únicamente conexiones con servicios de Firebase, como Authentication (autenticación de usuarios), Firestore (storage de datos) y Cloud Messaging (notificaciones).

#### Diagrama de conexiones externas a la app



*Fase de Diseño. 16 Conexiones aplicación*

#### 3.5.3.1 Servicios Firebase

Para escoger la opción de storage a utilizar en esta app se hizo una comparativa de varios aspectos entre: MySQL/MariaDB, Realtime y Firestore.

Se tuvo en cuenta que esta aplicación no se espera que tenga muchas conexiones simultáneas, escrituras, lecturas, ni volúmenes de datos grandes almacenados, que la

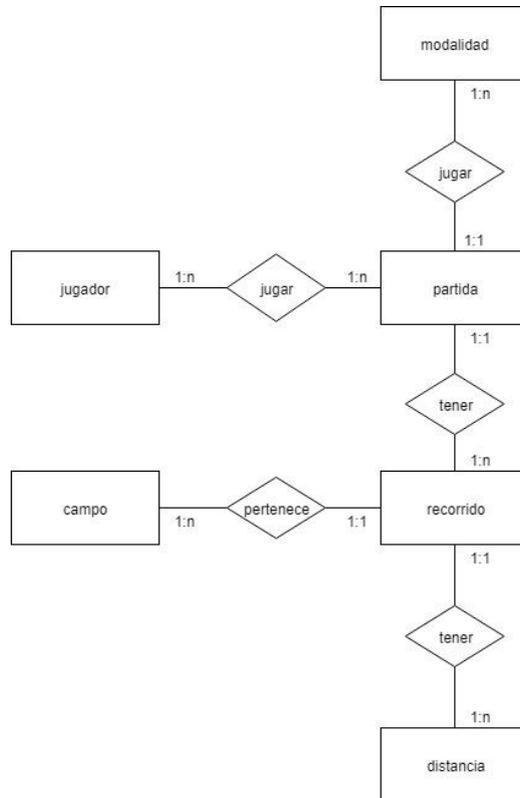
aplicación deberá funciona con los datos descargados en local, y solo escribirá los resultados una vez finalizada la partida, y que debe sincronizar los datos una vez tenga conectividad.

Después de realizar esta tabla la opción escogida fué Firestore, es la más novedosa, tiene sincronización automática, está integrada en Google con los otros dos servicios a utilizar de autenticación y notificaciones, y que parece que puede tener más futuro que Realtime, asumiendo que los jugadores de golf son usuarios que tienen todos una version alta de Android.

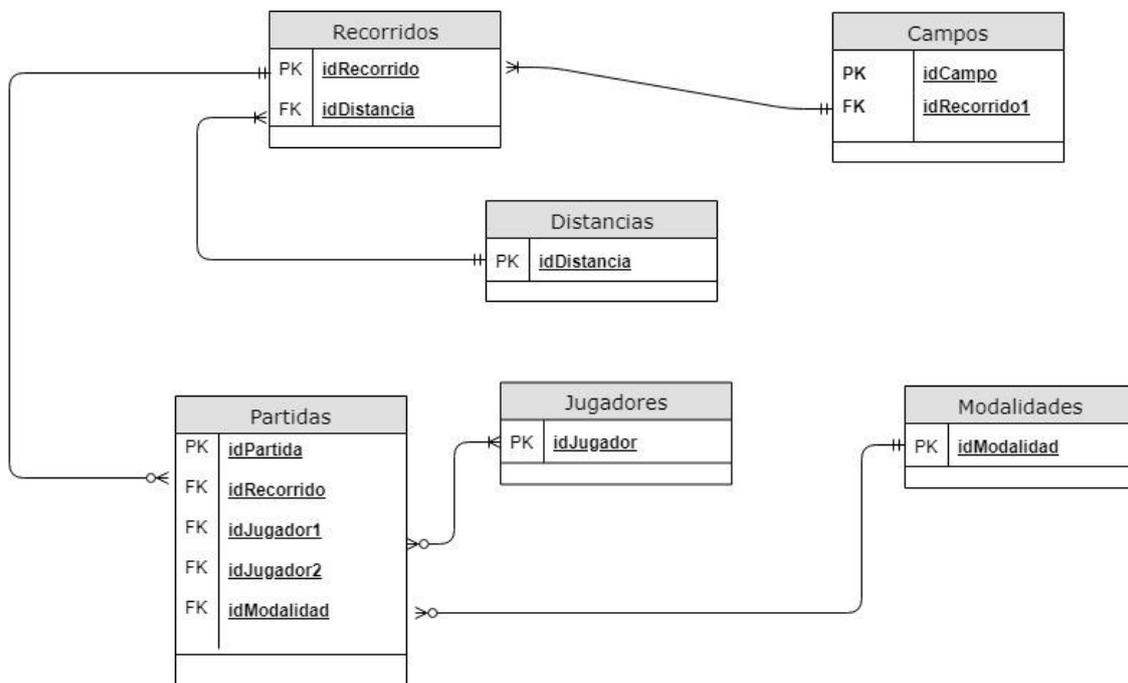
Concepto	mySQL/MariaDB	Realtime (plan Spark)	Firestore (plan Spark)
Tipo base de datos	Base de datos SQL.	noSQL	noSQL
Datos jerarquicos	Buena para datos jerarquicos	Buena para datos simples	Buena para datos jerarquicos
Rendimiento	Rendimiento medio/bajo	Rendimiento medio	Rendimiento bueno
Sincronizacion todos dispositivos	Hay que implementarla	API sincronizacion a todos los dispositivos	API sincronizacion a todos los dispositivos
Coste	Tiene coste	Gratis 1GB storage	Gratis 1GB storage Lecturas : 50K Escrituras: 20K
Modo offline	Hay que implementar la descarga de datos		Implementada en Firestore automaticamente
Version Android	Cualquiera		Android 4.0
Autenticacion	Hay que implementarla		API Authentication
Notificaciones	Hay que implementarlas		API Cloud messaging
Tipos de datos	Para guardar imagenes y ficheros mejor guardarlos en un webservice		Incluye hosting para cualquier tipo de datos en la funcionalidad de Cloud Storage for Firestore.

### 3.5.4 Diagrama entidad-relación (datos)

Incluyo dos representaciones en formatos distintos de como se relacionan las entidades para una mejor percepción.



Fase de Diseño. 17 Diagrama Entidad - Relacion (I)



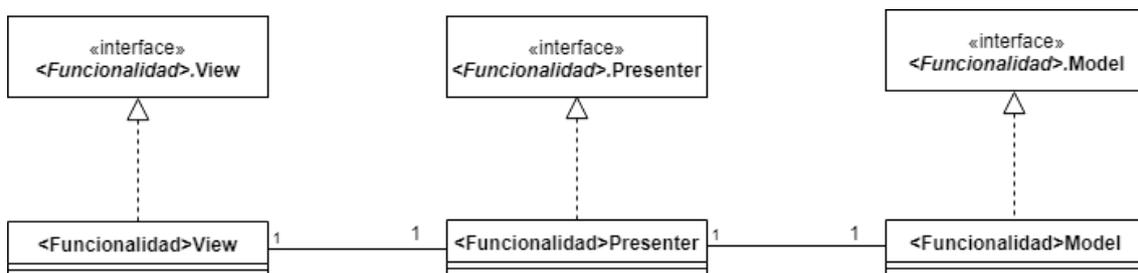
Fase de Diseño. 18 Diagrama Entidad - Relacion (II)

### 3.5.5 Diagramas de clases

En esta sección incluyo dos diagramas de clases, uno para representar el modelo MVP de las funcionalidades-activities, y el segundo para representar las entidades/objetos que maneja la aplicación.

#### **Diagrama de clases para las funcionalidades/activities del modelo MVP:**

Se creará un fichero interface por funcionalidad/activity, y dentro se declararán 3 interfaces, una para cada tier del paradigma MVP (view, presenter, model), que incluirán las funcionalidades para cada tier. De estas interfaces se implementarán cada una de las clases del paradigma MVP.

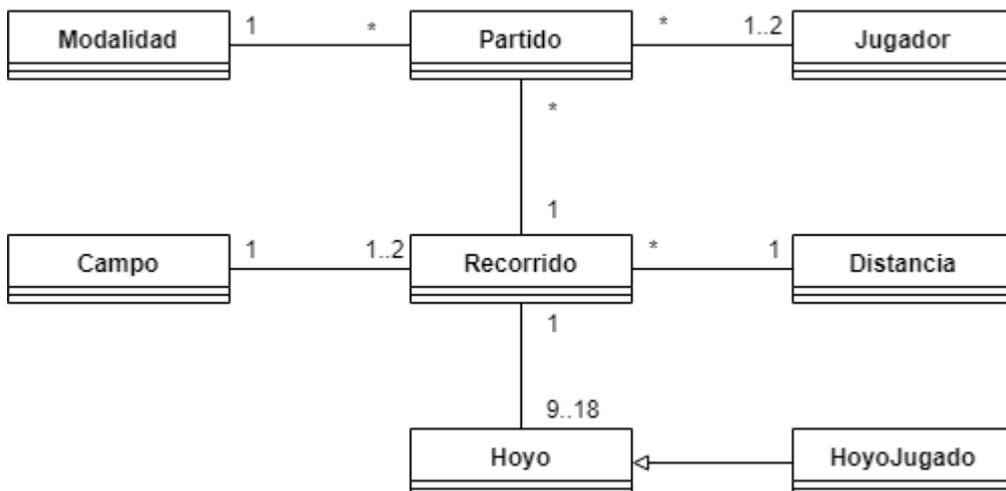


Fase de Diseño. 19 Diagrama de clases activities

#### **Diagrama de clases para las entidades/objetos que maneja la app:**

En este diagrama he incluido solo las cajas con el nombre de las entidades para facilitar la claridad de la representación, ya que algunas de ellas tienen muchos atributos y métodos (como la clase Campo). Por este motivo, los atributos y métodos se indican a continuación de los diagramas en unas tablas.

Finalmente no todas estas entidades lógicas ha sido necesario implementarlas como clases entidades físicas, ya que por ejemplo Distancia, solo hay 2 posibles valores que han sido declarados como constantes.



Fase de Diseño. 20 Diagrama de clases entidades

**Clases : Atributos & Métodos**

Modalidad	Distancia
-id:number -descripcion:string	-id:number -descripcion:string
+getId():number +getDescripcion():string	+getId():number +getDescripcion():string

HoyoJugado	Recorrido
-golpesJ1: number -golpesJ2: number -puntosJ1: number -puntosJ2: number	-id: number -idCampo: number -numeroHoyos: number -listaHoyos: Array<Hoyo>
+setGolpesJ1(number) +setGolpesJ2(number) +setPuntosJ1(number) +setPuntosJ2(number)	+getId(): number +getIdCampo(): number +getNumeroHoyos(): number +getListaHoyos(): Array<Hoyo>
+getGolpesJ1():number +getGolpesJ2():number +getPuntosJ1():number +getPuntosJ2():number	
Hoyo	Campo
-idRecorrido: number -numeroHoyo: number -metros: number -hcp: decimal -par: number -bunkers: number -agua: number -geoGreen: string -detalles: string	-id: number -numeroRecorridos: number -nombre: string -telefono: string -direccion: string -geoLocalizacion: string -url: string -email: string -logo: string -foto: string -bar: bool -vestuarios: bool -puestosPracticas: number
+getIdRecorrido(): number +getNumeroHoyo(): number +getMetros(): number +getHcp(): decimal +getPar(): number +getBunkers(): number	+getId(): number +getNumeroRecorridos(): number +getNombre(): string +getTelefono(): string +getDireccion(): string +getGeoLocalizacion(): string +getUrl(): string +getEmail(): string +getLogo(): string +getFoto(): string

+getAgua(): number +getGeoGreen(): string +getDetalles(): string	+getBar(): bool +getVestuarios(): bool +getPuestosPracticas: number
--	---

Jugador
-licencia:string -nombre:string -hcp:decimal -pin:number -foto:string -email: string
+setLicencia(string) +setNombre(string) +setHcp(decimal) +setPin(number) +setFoto(string) +setEmail(string)
+getLicencia():string +getNombre():string +getHcp():decimal +getPin():number +getFoto():string +getEmail(): string

Partido
-idRecorrido: number -idModalidad: number -fecha: date -jugador1: Jugador -jugador2: Jugador -teeInicial: number -resultados: Array<HoyoJugado> -puntosJ1Scratch: number -puntosJ2Scratch: number -puntosJ1Hcp:number -puntosJ2Hcp:number
+setIdRecorrido(number) +setIdModalidad(number) +setFecha(date) +setJugador1(Jugador) +setJugador2(Jugador) +setTeeInicial(number) +setResultados(Array<HoyoJugado>) +setPuntosJ1Scratch(number) +setPuntosJ2Scratch(number) +setPuntosJ1Hcp(number) +setPuntosJ2Hcp(number)
+getIdRecorrido(): number +getIdModalidad(): number +getFecha(): date +getJugador1(): Jugador +getJugador2(): Jugador +getTeeInicial(): number +getResultados(): Array<HoyoJugado> +getPuntosJ1Scratch():number +getPuntosJ2Scratch():number +getPuntosJ1Hcp():number +getPuntosJ2Hcp():number

## 4 Evaluación y correcciones del diseño

He realizado los siguientes cambios en el diseño propuesto inicialmente, a medida que he ido avanzando en la implementación y conociendo un poco más en detalle las API de Google:

**Autenticación:** Inicialmente propuse que fuera por usuario y password, a posteriori lo he cambiado a email y password, para adecuarlo al API de autenticación de Firebase.

**Menú Navigation Drawer:** He incluido el email del usuario como un dato nuevo a mostrar.

**Usabilidad & Intents implícitos:** Inicialmente puse campos como URL, email o telefono como campos solo de lectura. Pero durante la implementación de los layouts decidí cambiarlos por botones y si el usuario hace tap sobre el botón URL, teléfono o email del campo, se lanzará una acción para abrir la aplicación implícita de abrir un navegador, hacer una llamada, o abrir el cliente de correo.

**Arquitectura de conexiones y servicios:** He incluido un nuevo servicio que será el Firebase Storage, para guardar y servir las imágenes de la app.

**Multi-idioma:** He incluido una sección explicando el porqué del multi-idioma de la app en la sección Estrategia de contenido.

**Layouts:** Para mejorar la performance de renderización he utilizado layouts mas simples como los LinearLayout, re-colocando algunos elementos de las pantallas, con lo que es posible no quede exactamente igual que en los wireframes.

## 5 Implementación

En esta sección se detallan y explican los aspectos más relevantes de la implementación realizada y las motivaciones de escoger una u otra solución.

### 5.1 Modelo MVP

Como se ha explicado en la sección de diseño, se ha seguido un paradigma MVP (Model-View-Presenter), para de esta manera conseguir separar más claramente el front-end (views) del backend (models), así en un futuro si se quiere cambiar el backend por otro se podrá realizar de una forma más limpia y transparente para las views y presenters modificando únicamente las clases models.

De esta forma el tier de las views solo se comunica con el tier presenter al igual que el tier model. El tier presenter se comunica con ambas, haciendo de gestor de peticiones que llegan desde la view y gestor de datos a entregar a la view que llegan como respuesta desde el model (*Fig 15*).

Para implementar la app siguiendo este paradigma, se ha hecho lo siguiente para cada una de las activity, si cogemos la activity de Login como ejemplo:

**Interface:** Se crea una interface por cada una de las activity. Esta interface contiene 3 interface internas, una para cada tier, Model, View, Presenter. Cada una de estas interfaces contiene los métodos de interacción entre ellas. Ejemplo del Login:

```
public interface Login {  
  
    interface View{  
        void showResult(boolean result, String msg);  
    }  
  
    interface Presenter {  
        void authenticatePlayer(String email, String password, Activity actividad);  
        void validationFinished(boolean result, String msg);  
    }  
  
    interface Model{  
        void authenticatePlayer(String email, String password, Activity actividad);  
        void validationFinished(boolean result, String msg);  
    }  
  
}
```

**Clases:** Se construye una clase por cada tier. Ejemplo de Login:

[LoginView.java](#) ←(clase que extiende de Activity)  
[LoginModel.java](#)  
[LoginPresenter.java](#)

Y cada una de estas clases implementará cada una de las interfaces internas definidas anteriormente, implementando los métodos de cada interface. Ejemplo del Login:

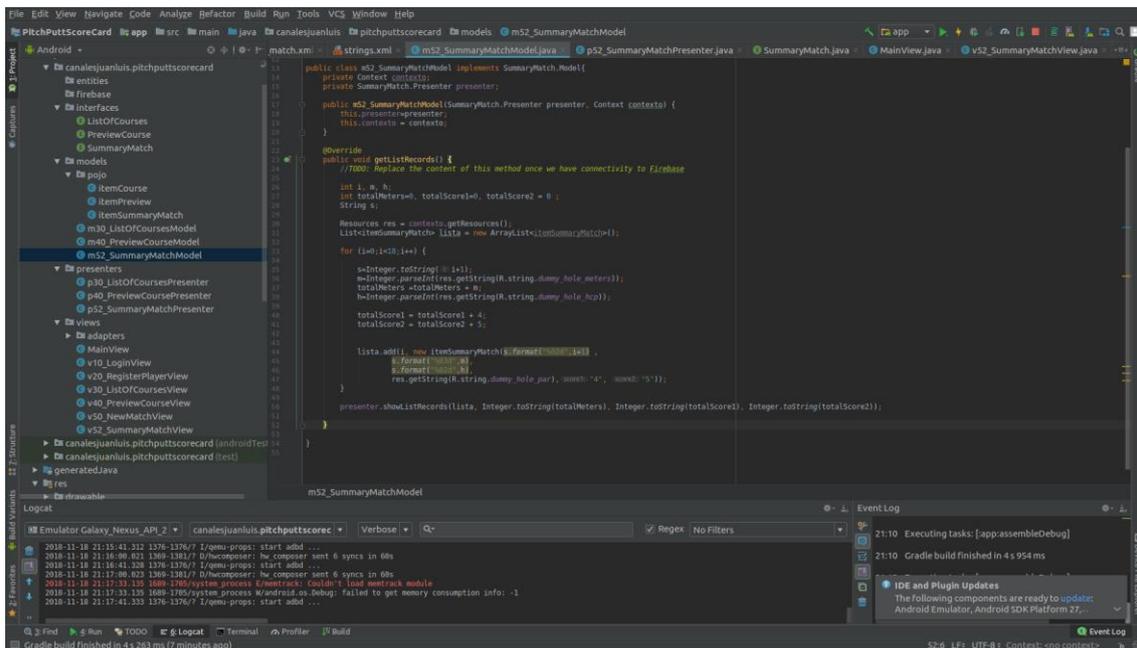
```
public class LoginView extends AppCompatActivity implements Login.View
public class LoginPresenter implements Login.Presenter
public class LoginModel implements Login.Model
```

## 5.2 Estructura del proyecto

El proyecto está estructurado como se indicaba en la sección de diseño, y como se muestra en este screenshot en los paquetes indicados de: **entities**, **firebase**, **interfaces**, **models**, **presenters** y **views**. Hay otro paquete adicional de **utilities**, donde he incluido clases para constantes, variables del sistema y validaciones

Dentro de models hay otro subpaquete llamado POJO, donde se incluyen las clases de los objetos más simples (con sus getters/setters) que utilizan los modelos para crear las listas a devolver a los presenters.

Y dentro de views hay otro subpaquete llamado adapters, para incluir las clases con los adapters que utilizan los recyclersviews.



## 5.2.1 Nomenclatura

### Activities

Como esta aplicación contiene muchos layouts, los he numerado con un sufijo numerado por delante para una mejor y más rápida identificación, al así estar ordenados, y seguido el mismo criterio también con los elementos que contienen, para identificarlos mejor dentro del conjunto de elementos en el fichero R. Todos los layouts y elementos visuales que contienen empiezan con la nomenclatura : vXX\_<nombre\_pantalla/nombre\_elemento> , donde XX es una numeración.

Para las clases vistas, modelo y presenter he seguido la misma numeración, para tener todas las clases ordenadas bajo el mismo criterio, en los diferentes paquetes, e identificarlas más fácilmente:

Vistas: vXX\_<nombre>  
Modelos: mXX\_<nombre>  
Presenters: pXX\_<nombre>

### Elementos visuales

Para los elementos de las pantallas, he seguido la siguiente nomenclatura para estandarizar los nombres e identificar rápido en el código de qué tipo de control visual se trata:

ImageViews: img<Nombre>  
TextViews: txt<Nombre>  
EditTexts: edt<Nombre>  
Buttons: btn<Nombre>  
Spinners: spn<Nombre>  
ToggleButton: tbn<Nombre>

### Resources

#### *strings.xml*

En el fichero strings se han incluido todos los textos de las UI: textos en controles y mensajes de usuario.

#### *colors.xml*

En este fichero se han configurado algunos colores aparte de los de defecto.

#### *dimens.xml*

Se han definido algunas dimensiones, sobre todo para igualar separaciones y márgenes entre controles, y standarizar diferentes tamaños de letra.

## 5.3 Views

### 5.3.1 Layouts

El criterio para escoger los layouts para cada una de las pantallas ha sido utilizar el layout más sencillo de renderizar en base a como quiero organizar los elementos visuales en cada una de ellas. En lugar de utilizar un ConstraintLayout para todas ellas he preferido utilizar otros Layouts sin constraints fijas para cada elemento.

- **Main:** LinearLayout. Orientation : vertical
- **Login:** LinearLayout. Orientation: vertical (contendran los campos usuario y password).  
LinearLayout. Orientation: horizontal (contendra los dos botones)
- **Registro Jugador:** LinearLayout. Orientation: vertical
- **Listado de Campos:** LinearLayout & RecyclerView
- **Ficha de Campo:** TableLayout
- **Preview Recorrido:** LinearLayout & RecyclerView
- **Detalle Hoyo X:** LinearLayout. Orientation:Vertical  
LinearLayout. Orientation: Horizontal (metros , par Hcp)
- **Nueva partida:** TableLayout.
- **Hoyo en juego:** TableLayout
- **Resumen Partida:** TableLayout
- **Historial Resultados:** LinearLayout & RecyclerView
- **Partida Historial:** TableLayout
- **Jugador Oponente:** LinearLayout. Orientation:vertical

## 5.4 Models

### 5.4.1 POJOs

En este paquete he incluido las clases simples que contienen los datos de los item de los distintos recyclerviews utilizados en cuatro views:

POJO	View donde se utiliza
itemCourse	ListOfCourses
itemPreview	PreviewCourse
itemRecordsResults	RecordsResults
itemSummaryMatch	SummaryMatch

Estos POJO, se informan en los modelos de cada una de las activities.

## 5.5 Presenters

Los presenters son meras clases *pasarela* que implementan cada una de las interfaces definidas para cada activity, y hacen de pipeline entre las peticiones que llegan de las views se le pasan a los modelos, y las respuestas que llegan de los modelos se les pasan a las views.

La función de esta capa adicional es meramente la de aislar la capa de interface de usuario del backend que esté conectado, mediante un nivel más de abstracción, que implementa una interface sin necesidad de tener conocimiento el backend utilizado.

Si en un futuro se decide cambiar el backend, habría que modificar modelos, y sustituir las clases de firebase, manteniendo el front-end y presenters con la misma interface.

## 5.6 Entities

Este paquete contiene las clases que representan las entidades que maneja la app, como son: Player, Match, Course, Hole, y algunas adicionales que he tenido que implementar como PlayerWithPhoto (incluye la clase Player y un Bitmap para la foto) & HolePlayed.

Algunas de estas clases he tenido que modificarlas respecto al diseño original, después de ver las limitaciones en los tipos de datos soportados por Firestore, por ejemplo Firestore no soporta bitmaps, o arrays de bytes, por tanto la foto de los jugadores no podía grabarla en Firestore, sino en Cloud Storage.

## 5.7 Firebase

Para el manejo de Firebase he utilizado una única clase que contiene los métodos necesarios para el manejo de los módulos: ***Authentication, Firestore, Cloud Storage***

Para la utilización de los servicios de Firebase cuyos métodos finales al ser operaciones de I/O (authentication, firestore y storage) utilizan llamadas asíncronas, se han tenido que definir una serie de interfaces con métodos onCallBack, para recuperar los datos de las diferentes llamadas asíncronas una vez van finalizando, pasando como parámetros los datos recuperados en esta llamada asíncrona. Uno de los parámetros de estos métodos era un objeto del tipo de esta interface.

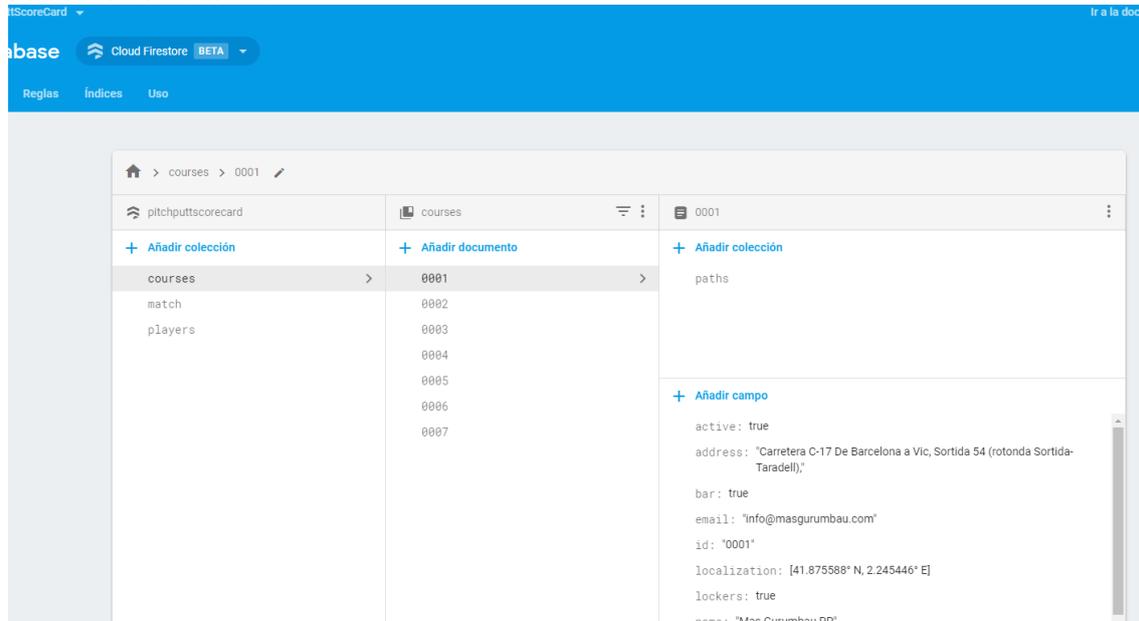
Los métodos más destacados de esta clase son:

Funcionalidad	Servicios utilizados	Método
Logout del módulo de autenticación	Authentication	public static void logoutFirebase
Login en módulo de autenticación (mediante email & password)	Authentication	public static void userAuthenticationFirebase
Registro nuevo usuario	Authentication (email & password) Firestore (resto de datos de usuario) Cloud Storage (foto de usuario)	public static void userNewRegisterFirebase
Recupera imagen	Cloud Storage	public static void getImagefromStorage
Recupera datos de una collection	Firestore	public static void getCollection

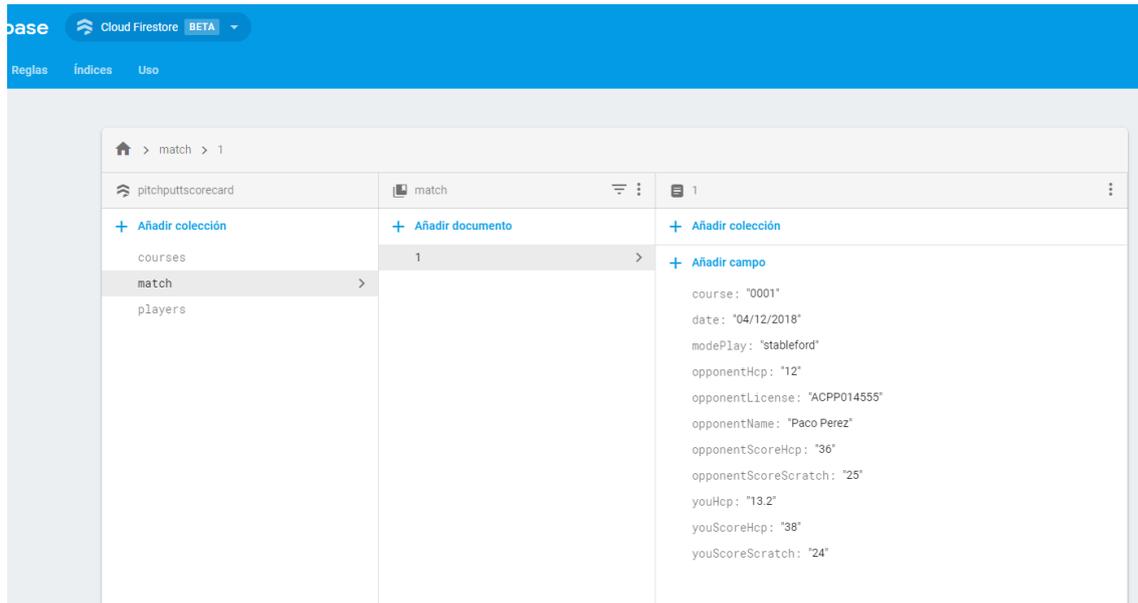
## 5.7.1 Estructura de datos en Firebase

En estas capturas de imágenes se muestra la estructura de los datos y configuración que finalmente se ha hecho en firebase.

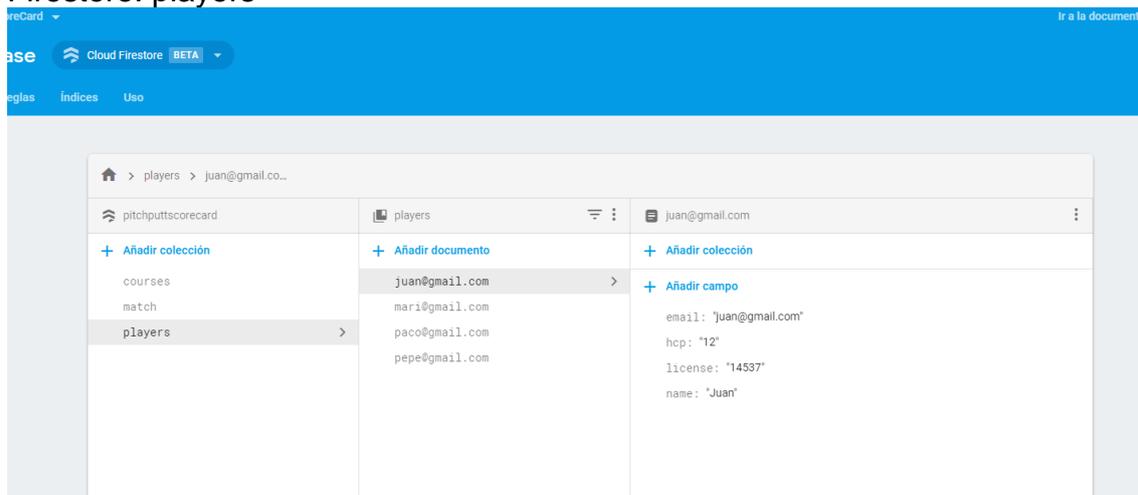
### Firestore : courses



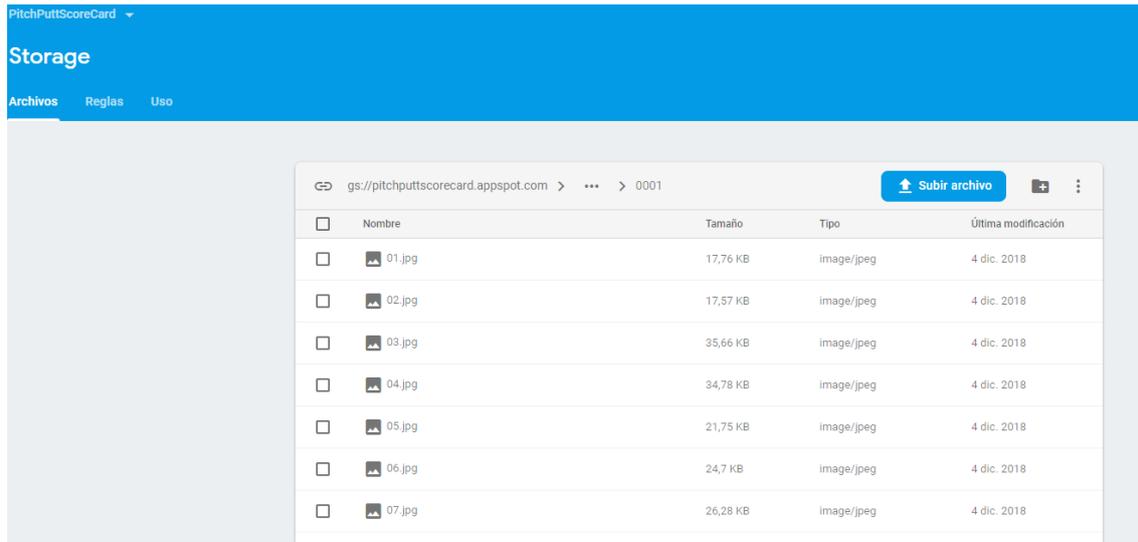
## Firestore: match



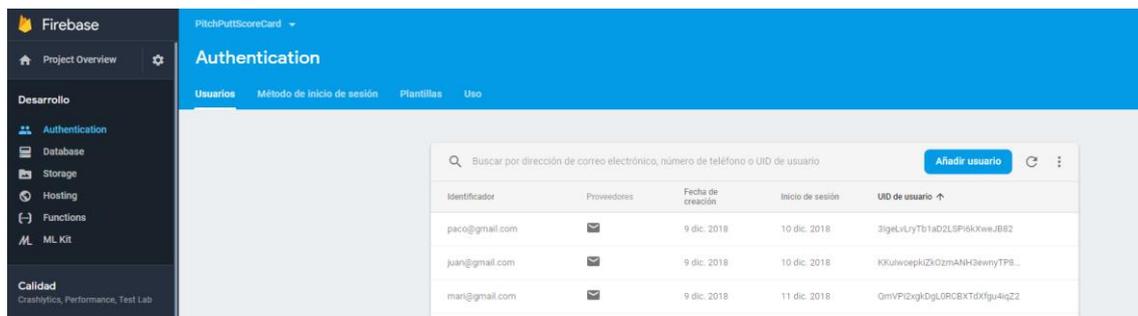
## Firestore: players



## Storage: Imágenes de los campos



## Authentication



## 5.7.2 Latencia & Performance

Todos los datos de campos, datos de jugadores, datos de partidos están almacenados en Firebase Firestore, y las imágenes en Cloud Storage.

Cada campo tiene 21 imágenes que hay que cargar mediante llamadas asíncronas al servicio de Storage, y a pesar que he reducido las imágenes a unos pocos Kbytes cada una, lo que veo que más penaliza es la latencia de las llamadas asíncronas al servidor. No se si es debido a que estoy utilizando el servicio gratuito, pero es bastante tiempo el que lleva en cargar una imagen.

Todas las imágenes de un campo no llegan a ocupar de media 500Kbytes, por tanto el problema no es el tamaño, sino la latencia de múltiples llamadas en cada pantalla, por lo que para mejorar la performance de la app y la usabilidad y evitar tener que hacer continuas llamadas asíncronas durante el juego al servidor Storage lo que he hecho es al iniciar la app, cargar todas las imágenes de los campos en una estructura en memoria.

Esta carga se realiza al iniciar la app donde aparecerá una pantalla de inicio de la app y un progress bar con el texto “Loading data...”, es más aceptable que el usuario

tenga que esperar 10seg al iniciar una app si se le informa que esta cargando datos, que tener que sufrir lentitud en llamadas asíncronas durante todo el uso de la app y ver como las imágenes en cada pantalla se van cargando una a una.

Esta carga de datos iniciales en memoria en las pruebas que he hecho tarda de media menos de 10 segundos, y a partir de ahí con los datos en memoria en el manejo de pantallas con imágenes no se notan delays o que las pantallas vayan cargando poco a poco.

En un futuro como mejora, si la aplicación tuviera varias decenas o centenas de campos mapeados, no sería necesario tampoco cargar todos en memoria pero lo que sí se podría plantear es solo cargar los 2 o 3 campos más cercanos, mirando la localización gps de los campos y posición gps actual del móvil.

## 6 Valoración económica del trabajo

### 6.1 Gastos y Beneficios

Los gastos para hacer este prototipo inicial para el propósito de trabajo de fin de máster, han sido únicamente las horas de programación, ya que las imágenes se han buscado de librerías de imágenes de uso libre, algunas imágenes (logos de campos) se han utilizado la de los de sites públicos de los campos, y los servicios de Firebase se ha escogido el Spark Plan que es gratuito.

Pero si se quisiera poner esta app en el mercado habría que hacer algunos pasos adicionales, y que algunos de ellos conllevarían gastos :

- Comunicar a los campos a incluir, para informarles que seran incluidos con sus logos e información oficial que se muestre en el site.
- Añadir más imágenes a la app para hacerla más atractiva. Alguna de ellas podría ser bajo pago de derechos.
- Si se desea implementar la app para iOS y subirla al Apple Store, habría que contar con el gasto de suscripción anual de desarrollador, y la compra de un MacPro.
- Utilizar un plan de servicio de Firebase más avanzado dependiendo del volúmen de almacenamiento y de trafico habría que ver cual es el plan a contratar. Teniendo en cuenta los precios : <https://firebase.google.com/pricing/>

### 6.2 Viabilidad del producto

Aún teniendo en cuenta los gastos antes mencionados para comercialitzar la app, creo que si tendría viabilidad teniendo en cuenta la cantidad de licencias que hay actualmente , y que està creciendo año tras año.

Solo en Catalunya hay aprox 36 campos de pitch & putt, y 15.000 licencias de jugadores de pitch and putt (sin contar las licencias de golf) , y creciendo año tras año.

Pienso que la mejor opción para este tipo de apps, para poder subvencionar el incluir nuevos campos y funcionalidades es ofrecer una versión gratuita con un número de campos y funcionalidades limitados, y ofrecer un upgrade a una versión Premium con un pago de suscripción anual.

Teniendo en cuenta que una de las apps del mercado està cobrando 49 EUR anuales, creo que sería viable empezar con una suscripción anual de entre 4 o 5 EUR anuales, para ir cubriendo las horas de trabajo para tenerla actualizada. Mejor que plantear un enfoque de negocio de influir anuncios, que molestan al jugador que necesita apuntar rápido y ver la info del hoyo sin tener que esperar a que acabe el anuncio.

## 7 Conclusiones

### Lecciones aprendidas

Como lecciones aprendidas, aparte de aprender temas técnicos concretos de la programación en apps Android, y el uso de los Servicios de Firebase, lo que quizás he aprendido es a valorar mucho más todo el trabajo y codificación que hay detrás de una pequeña app, y que aún habiendo potentes librerías con múltiples métodos hay que codificar más de lo que parece, utilizar muchas horas en aprender lo más que puedas sobre todas las librerías y APIs que hay, y sobre todo tener en cuenta múltiples aspectos que en otro tipo de programación no son tan necesarios, como el tema de optimización de espacio y uso de recursos, y controlar la programación de tareas asíncronas y multithreading sobre todo con el uso de las API de Firebase, que solo conocía muy por encima. Al desconocer realmente como utilizarlas bien, algunas limitaciones que me he ido encontrando, no calculé correctamente en la fase de diseño, la valoración de tiempo a dedicarle, y no calculé bien el tiempo real de la curva de aprendizaje que finalmente he tenido que emplear, no solo para Firebase.

### Logro de objetivos

Inicialmente pensaba que me iba a ser más sencillo implementar esta app, pero como he comentado en el párrafo anterior el uso de las API de Firebase, me ha consumido mucho más tiempo del planificado, y he tenido que recortar algunos aspectos que me hubiera gustado que hubieran estado más trabajados como por ejemplo enriquecer y cuidar más los detalles del diseño de la app, o incluir funcionalidades que me iban viniendo a la mente como las que indico en el apartado de líneas pendientes y que finalmente no he tenido tiempo de incluir.

Los objetivos principales definidos en el diseño inicial he podido conseguir finalizarlos pero sin poder realizar ninguna de las mejoras adicionales pensadas, y he tenido que recortar alguna pantalla de las menos importantes, a pesar de tener que invertir más horas de las que había estimado.

### Metodología y Planificación

La metodología creo que sí ha sido la correcta, he intentado dentro de la fase de implementación seguir las siguientes subfases:

- a. Crear las activities sin diseño e implementar la navegación entre ellas
- b. Diseño de las activities sin código
- c. Crear las interfaces para cada una de las activities (modelo MVP)
- d. Crear las clases presenters y models
- e. Integrar con backend (Firebase)
- f. Mejoras: Mejorar diseño visual app & Añadir funcionalidades adicionales

Pero como he comentado en los párrafos anteriores, el problema ha venido sobre todo en la penúltima subfase cuando he llegado a la integración con el backend, Firebase, porque además de ser la última fase importante, y ya no tenía mucho tiempo de reacción, con lo que he tenido que hacer más sobreesfuerzos de lo planificado para cubrir el gap de desconocimiento tecnológico.

Y reconozco que además del gap tecnológico, fui demasiado optimista en el diseño de funcionalidades para el tiempo disponible que tenía y mi rapidez implementando, lo que finalmente no he podido acabar la app en el tiempo estimado, a pesar de la gran cantidad de horas. Había estimado la fase de implementación terminarla en 125h, y finalmente llevo registradas más de 140h y sin conseguir añadir ninguna de las mejoras pensadas.

### **Lineas pendientes de trabajo**

Mejorar el diseño de la app para hacerla más atractiva, definiendo un Thema con styles personalizados.

Incluir un widget de previsión de tiempo meteorológico en los diferentes campos

Recomendaciones de palos y tipo de golpe, según la distancia y orografía del hoyo / green.

Incluir una geolocalización de que campos tienes más cercanos respecto a tu posición actual, si el usuario está de vacaciones por ejemplo y no conoce la zona.

Incluir otros idiomas

Ir incluyendo más campos no solo cercanos, sino Internacionales

Monetizar la app tal como se ha indicado en el apartado de Viabilidad del producto.

Implementar la app para iOS

Crear un site web donde se pueda crear una comunidad de jugadores y puedan compartir resultados, retarse en puntuaciones, estadísticas, blog con recomendaciones entre ellos, etc...

Finalizar las funcionalidades del diseño inicial que se quedaron pendientes

## 8 Glosario

Término	Descripción
HCP / Handicap	Nivel / ventaja en golpes de un jugador. O dificultad en un hoyo
Par	Número de golpes que se ha establecido para finalizar un hoyo
Scratch	Puntuación bruta sin tener en cuenta handicap

## 9 Derechos de imagen

Los logos , croquis de campos y algunos datos públicos de los campos incluidos en esta app se han obtenido de la web oficial de pitch and putt en Cataluña [www.pitch.cat](http://www.pitch.cat), e información pública de los sites públicos de los campos incluidos.

Y para el diseño de la app se han utilizado las siguientes imágenes de librerías de imágenes libres de derechos de autor:

Uso	Link a imagen / site	License
FirstWelcome activity	<a href="https://pixabay.com/es/golf-puesta-del-sol-deporte-787826/">https://pixabay.com/es/golf-puesta-del-sol-deporte-787826/</a>	CC0 License ✓ Free for personal and commercial use
Menu main	<a href="https://pixabay.com/es/golf-verde-fairway-bosque-%C3%A1rboles-2158897/">https://pixabay.com/es/golf-verde-fairway-bosque-%C3%A1rboles-2158897/</a>	✓ No attribution required
Menu main	<a href="https://pixabay.com/es/golf-club-de-golf-solamente-verde-1208900/">https://pixabay.com/es/golf-club-de-golf-solamente-verde-1208900/</a>	

## 10 Bibliografía

Ref	Web / Libro	Visitada
1	<a href="https://www.golfsitges.com/modalidades-de-juego">https://www.golfsitges.com/modalidades-de-juego</a>	30/10/2018

Aparte de la referencias indicadas arriba de consulta más específica, quiero también agradecer los recursos de aprendizaje que he ido encontrando en las siguientes webs, que me han ayudado a entender diferentes conceptos útiles para hacer este trabajo

Contenido	Website	Autor
Documentación oficial de Firebase	<a href="https://www.firebase.com/docs/">https://www.firebase.com/docs/</a>	Google
Documentación oficial de Android	<a href="https://developer.android.com">https://developer.android.com</a>	Google
Diferentes conceptos de programación en Android	Youtube: <a href="#">Videotutoriales</a>	José Dimas Luján Castillo
Manejo de tareas asíncronas de Firebase	Youtube: <a href="#">Video</a>	Alex Mamo

## 11 Anexos

Los siguientes ficheros anexos se incluyen aparte en el mismo paquete comprimido que este documento :

- Plan del proyecto
- Manual de usuario