



ASIGNATURA: SOFTWARE DISPOSITIVOS MÓVILES

PROYECTO FINAL: FOOTMATCH



Definición de la aplicación

El sistema se llama FootMatch y así será referenciado en el resto del documento.

La aplicación propuesta es una plataforma de consulta de datos de fútbol. De este modo la inspiración surgió en base a unas de las aplicaciones más usadas por los amantes del fútbol, que es BeSoccer.

FootMatch proporcionará a los usuarios información detallada sobre partidos, equipos, ligas y estadísticas relacionadas con el mundo del fútbol. Permitirá a los usuarios acceder a datos en tiempo real de los partidos que se están jugando hoy, los partidos jugados ayer y los programados para los próximos días. Además, ofrecerá detalles específicos de cada partido, incluyendo resultados, equipos que juegan, hora de inicio, alineaciones, árbitros y estadísticas relevantes.

Los usuarios también podrán explorar las ligas más importantes, accediendo a la clasificación de equipos. Además, podrán obtener información detallada sobre cada equipo, incluyendo datos sobre el entrenador, estadio, plantilla, entre otros. La aplicación también proporcionará rankings específicos de una liga, como goleadores, asistentes y goles de penalti.

El único sistema externo con el que se comunica FootMatch es con la API football-data.org (<https://www.football-data.org/documentation/quickstart>).

Objetivos

Objetivo	Definición
Proporcionar Información Integral del Fútbol	Brindar a los amantes del fútbol una plataforma integral para acceder a información detallada sobre partidos, equipos, ligas y estadísticas.
Documentación	Documento siguiendo la plantilla dada para su posterior entrega
Trabajo en equipo	Continuar con el desarrollo de nuestras habilidades como miembros de equipo adquiridas en asignaturas previas. Dichas habilidades son claves para el correcto desarrollo de cualquier proyecto con más de un miembro.
Aprender nuevas tecnologías	Familiarizarnos con el entorno tecnológico utilizado. Especialmente con Android Studio, con el desarrollo para móviles, además de la conexión con una API para recoger datos. También fue un reto hacer los diseños de la aplicación con XML y el uso de Kotlin (que eran totalmente nuevos para nosotros)

Funcionalidad

Requisitos funcionales

Requisito Funcional	Definición
RFPartidos1.	<p>El sistema permitirá al usuario elegir el partido que quiere ver.</p> <ul style="list-style-type: none">• RFPartidos1.1. Los partidos se podrán ordenar por tres categorías.<ul style="list-style-type: none">○ RFPartidos1.1.1. Ayer.<ul style="list-style-type: none">▪ RFPartidos1.1.1.1. Se mostrarán en caso de que haya partidos.○ RFPartidos1.1.2. Hoy.<ul style="list-style-type: none">▪ RFPartidos1.1.2.1. Se mostrarán en caso de que haya partidos.○ RFPartidos1.1.3. Todos.<ul style="list-style-type: none">▪ RFPartidos1.1.1.1. Se mostrarán en caso de que haya partidos.
RFPartidos2.	<p>El sistema mostrará el partido seleccionado</p> <ul style="list-style-type: none">• RFPartidos2.1. El sistema mostrará los siguientes datos.<ul style="list-style-type: none">○ RFPartidos2.1.1 Liga.○ RFPartidos2.1.2. Logo de la liga.○ RFPartidos2.1.3. Escudo de los equipos.○ RFPartidos2.1.4. Nombre de los equipos.○ RFPartidos2.1.5. Nombre del estadio.○ RFPartidos2.1.6. Estado del partido.• RFPartidos2.2. El sistema permitirá al usuario elegir que quiere ver entre las siguientes características.<ul style="list-style-type: none">○ RFPartidos2.2.1. Estadísticas.<ul style="list-style-type: none">▪ RFPartidos2.2.1.1. Mostrará datos de los últimos enfrentamientos entre los equipos.○ RFPartidos2.2.2. Plantillas.<ul style="list-style-type: none">▪ RFPartidos.2.2.2.1. Mostrará las plantillas de los dos equipos○ RFPartidos2.2.3. Árbitros.<ul style="list-style-type: none">▪ RFPartidos2.2.3.1. Mostrará el árbitro del partido.
RFLigas1.	<p>El sistema permitirá al usuario ver la clasificación de las LIGAS_LISTA.</p> <ul style="list-style-type: none">• RFLigas1.1. LIGAS_LISTA serán:<ul style="list-style-type: none">○ RFLigas1.1.1. Liga española.○ RFLigas1.1.2. Premier League.○ RFLigas1.1.3. Serie A.○ RFLigas1.1.4. Bundesliga.
RFLigas2.	<p>El sistema permitirá al usuario acceder a la lista de líderes de LISTA_ESTADISTICAS de dicha liga</p> <ul style="list-style-type: none">• RFLigas2.1. LISTA_ESTADISTICAS serán los siguientes campos.

	<ul style="list-style-type: none"> ○ RFLigas2.1.1. Máximos goleadores. <ul style="list-style-type: none"> ▪ RFLigas2.1.1.1. Mostrará los siguientes datos. <ul style="list-style-type: none"> • RFLigas2.1.1.1.1. Nombre y apellidos. • RFLigas2.1.1.1.2. Equipo. • RFLigas2.1.1.1.3. Goles. • RFLigas2.1.1.1.4. Goles por partido. • RFLigas2.1.1.1.5. Partidos jugados. ○ RFLigas2.1.2. Máximos asistentes. <ul style="list-style-type: none"> ▪ RFLigas2.1.2.1. Mostrará los siguientes datos. <ul style="list-style-type: none"> • RFLigas2.1.2.1.1. Nombre y apellidos. • RFLigas2.1.2.1.2. Equipo. • RFLigas2.1.2.1.3. Asistencias. • RFLigas2.1.2.1.4. Partidos jugados. ○ RFLigas2.1.3. Máximos goleadores de penalti. <ul style="list-style-type: none"> ▪ RFLigas2.1.3.1. Mostrará los siguientes datos. <ul style="list-style-type: none"> • RFLigas2.1.3.1.1. Nombre y apellidos. • RFLigas2.1.3.1.2. Equipo. • RFLigas2.1.3.1.3. Goles de penalti. • RFLigas2.1.3.1.4. Partidos jugados.
RFEquipos1.	<p>El sistema permitirá al usuario los equipos de las LIGAS_LISTA.</p> <ul style="list-style-type: none"> • RFEquipos1.1. Se mostrará al usuario la siguiente información. <ul style="list-style-type: none"> ○ RFEquipos1.1.1. Información de la historia del equipo. ○ RFEquipos1.1.2. Información del entrenador del equipo. ○ RFEquipos1.1.3. Información de la plantilla del equipo. ○ RFEquipos1.1.4. Acceso a la web externa del equipo. <ul style="list-style-type: none"> ▪ RFEquipos1.1.4.1. Si existe una, en caso de no existir no.

Requisito no funcional	Definición
RNFPeticiones1.	Las peticiones a la API externa no podrán superar las 10 por minuto.
RNFDisponibilidad1.	El sistema deberá estar disponible en más de un 99% del tiempo.

Diseño de la aplicación

Diagrama de paquetes

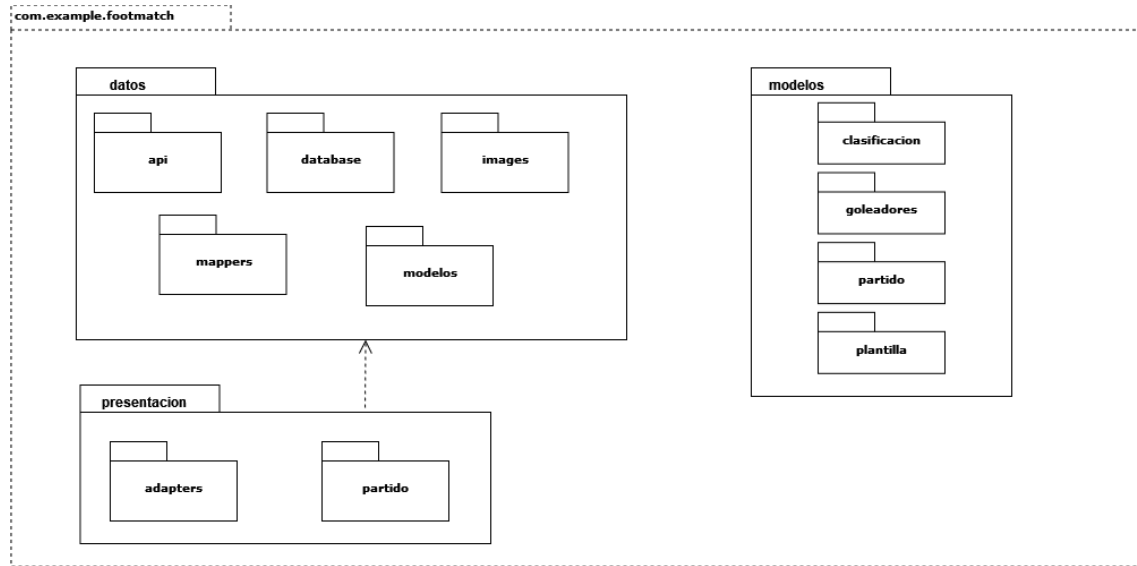
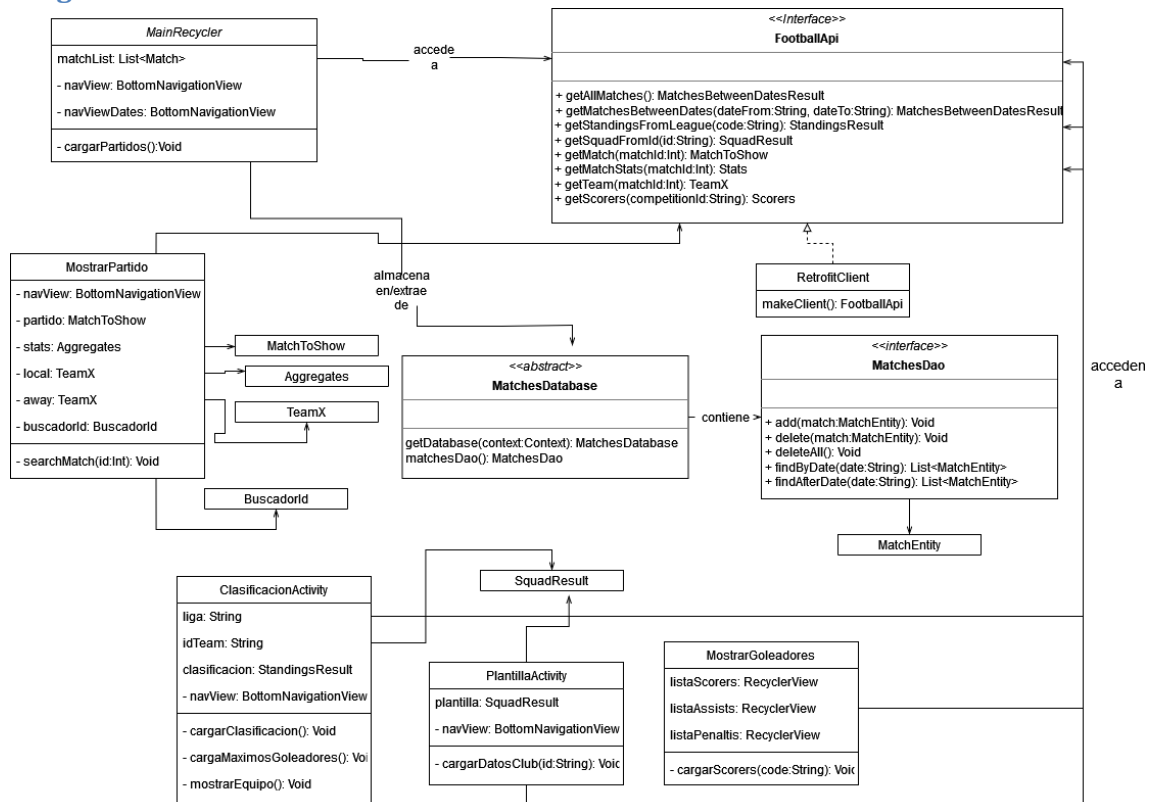


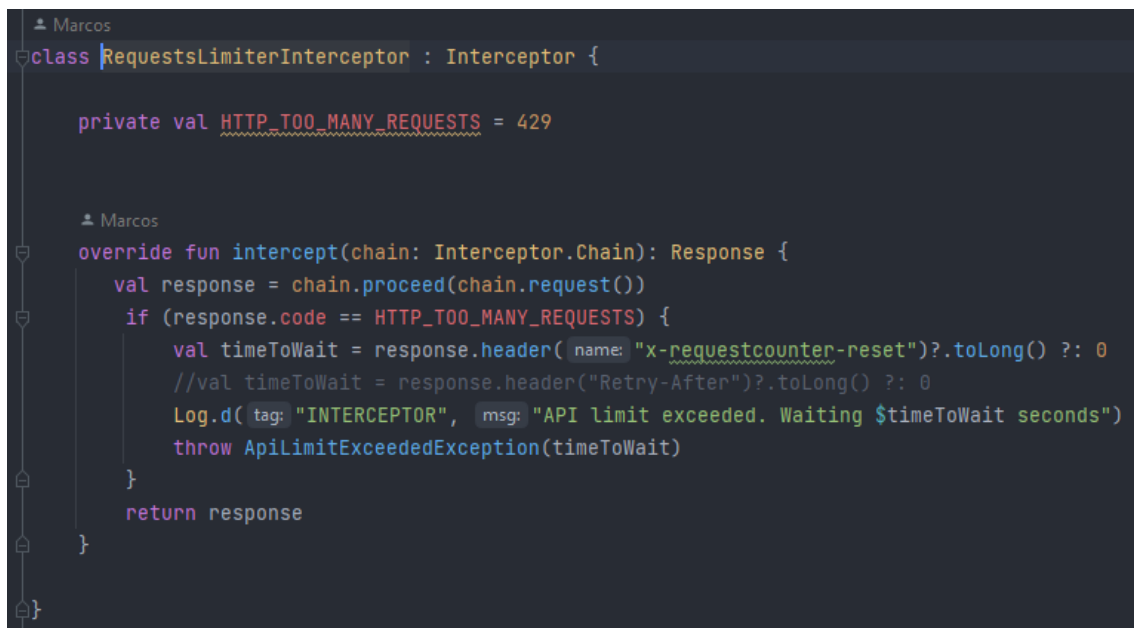
Diagrama de clases



Back-end

FootMatch obtiene los datos necesarios sobre partidos, equipos y ligas de una API Rest gratuita, denominada football-data.org. Cabe destacar que se ha empleado la versión 4 de dicha API. La uri base que se ha de emplear para realizar las consultas es <https://api.football-data.org/v4/>. Nótese que sin emplear una API key válida no se podrá realizar ninguna consulta. La documentación correspondiente a dicha versión de la API se encuentra disponible en <https://docs.football-data.org/general/v4/index.html>.

La capa gratuita de dicha API permite realizar 10 consultas/minuto. Esto ha llevado a implementar un control de las peticiones para que si se alcanza dicho límite se le indique al usuario el tiempo que tiene que esperar antes de poder realizar otra consulta.



```
class RequestsLimiterInterceptor : Interceptor {  
  
    private val HTTP_TOO_MANY_REQUESTS = 429  
  
    override fun intercept(chain: Interceptor.Chain): Response {  
        val response = chain.proceed(chain.request())  
        if (response.code == HTTP_TOO_MANY_REQUESTS) {  
            val timeToWait = response.header("x-requestcounter-reset")?.toLong() ?: 0  
            //val timeToWait = response.header("Retry-After")?.toLong() ?: 0  
            Log.d(tag: "INTERCEPTOR", msg: "API limit exceeded. Waiting $timeToWait seconds")  
            throw ApiLimitExceededException(timeToWait)  
        }  
        return response  
    }  
}
```

En la imagen superior se muestra el interceptor implementado que se ejecuta cada vez que se realiza una consulta a la API. De esta forma, si se alcanza el límite se lanza la excepción impidiendo que se lleve a cabo la consulta y permitiendo que se pueda mostrar al usuario el tiempo de espera necesario.

Además de la implementación del interceptor, para reducir el número de peticiones, se ha implementado una base de datos local que permite almacenar los partidos disputados el día anterior y los pendientes desde mañana. De esta forma, cuando el usuario selecciona una de dichas opciones en la pantalla principal o ambas, además de realizar la consulta a la base de datos para obtener los resultados, estos se añaden a la base de datos para que si se seleccionan en la misma ejecución de la app dichas opciones se recuperen los datos sin necesidad de acceder a la API.

```

└ Marcos
@Entity(tableName = "matches")
data class MatchEntity (

    @Embedded(prefix="away_")
    val awayTeam: AwayTeam,

    @Embedded(prefix="home_")
    val homeTeam: HomeTeam,

    @PrimaryKey(autoGenerate = false)
    val id: Int,

    val status: String,

    @Embedded(prefix="score_")
    val score: ScoreEntity,

    @ColumnInfo(name = "utc_date")
    val utcDate: String

)

```

En la imagen superior, se muestra la entidad empleada para almacenar la información necesaria de un partido, teniendo en cuenta que se trata de la pantalla principal “MainRecycler”.

A continuación, se muestran las consultas a la API empleadas por FootMatch para implementar toda la funcionalidad, que se detalla profundamente en el siguiente apartado.

```

└ Marcos
@GET("matches/")
suspend fun getMatchesBetweenDates(@Query("dateFrom") dateFrom: String,
                                     @Query("dateTo") dateTo: String): MatchesBetweenDatesResult

└ RichardPix12
@GET("competitions/{code}/standings")
suspend fun getStandingsFromLeague(
    @Path("code") code:String
):StandingsResult

└ RichardPix12
@GET("teams/{id}")
suspend fun getSquadFromId(
    @Path("id") id:String
):SquadResult

└ UO282337
@GET("matches/{matchId}")
suspend fun getMatch(@Path("matchId") matchId: Int): MatchToShow

└ UO282337
@GET("matches/{matchId}/head2head?limit=50")
suspend fun getMatchStats(@Path("matchId") matchId: Int): Stats

└ UO282337
@GET("teams/{teamId}")
suspend fun getTeam(@Path("teamId") matchId: Int): TeamX

└ UO282337
@GET("competitions/{code}/scorers")
suspend fun getScorers(@Path("code") competitionId: String): Scorers

```

Front-end

La aplicación tiene como objetivo el minimalismo y facilidad de uso. Es por ello, que en las pantallas se presentan el mínimo número de elementos necesarios para su uso. La navegación entre pantallas se realiza de tres formas: Pulsando el botón de atrás del dispositivo, pulsando el botón de inicio de la toolbar inferior (en aquellas pantallas que lo presentan) y mediante los botones de las diferentes pantallas de las ligas y equipos, que se explicarán a continuación.

Al iniciar la aplicación por primera vez se mostrará una activity donde se mostrarán un apartado con una serie de botones, que representan las ligas más importantes del mundo, donde haciendo click podemos acceder a ellas. También se muestran una lista de partidos que se jugarán ese mismo día, aunque contamos con una toolbar a través de la cual podemos acceder a los partidos que se jugaron ayer, y a los partidos que se jugarán en los próximos días (una semana a partir del día actual). Dicha activity, dispone también de la opción “swipe refresh”, a través de la cual el usuario puede actualizar los resultados obtenidos en cualquier momento, siempre que se encuentre seleccionada la opción “hoy”. Finalmente, tenemos un botón de inicio a través del cual podemos volver a esta pantalla de inicio (este botón estará disponible en casi todas las ventanas).

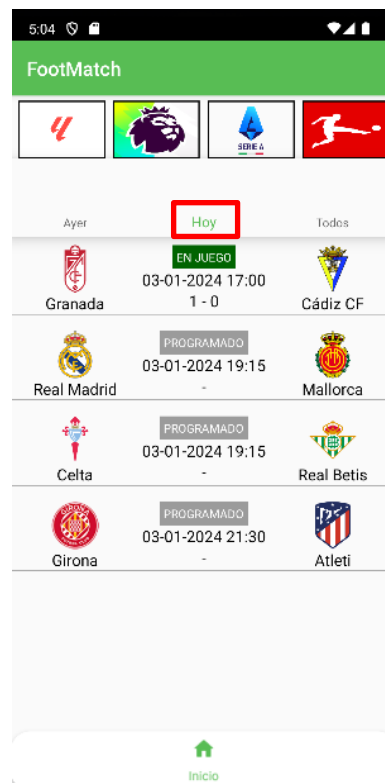


Figura 1. Pantalla (de Home) donde se listan los partidos de Hoy

A continuación, al seleccionar un partido cualquiera, pasaremos a otra activity en la que se mostrarán todos los detalles de este. Vamos a seleccionar el partido que está en juego, que en este caso es el Granada contra el Cádiz. Esta pantalla consta de tres fragmentos: uno para las estadísticas (aparece por defecto), otro para las plantillas de ambos equipos y otro para los árbitros del partido.

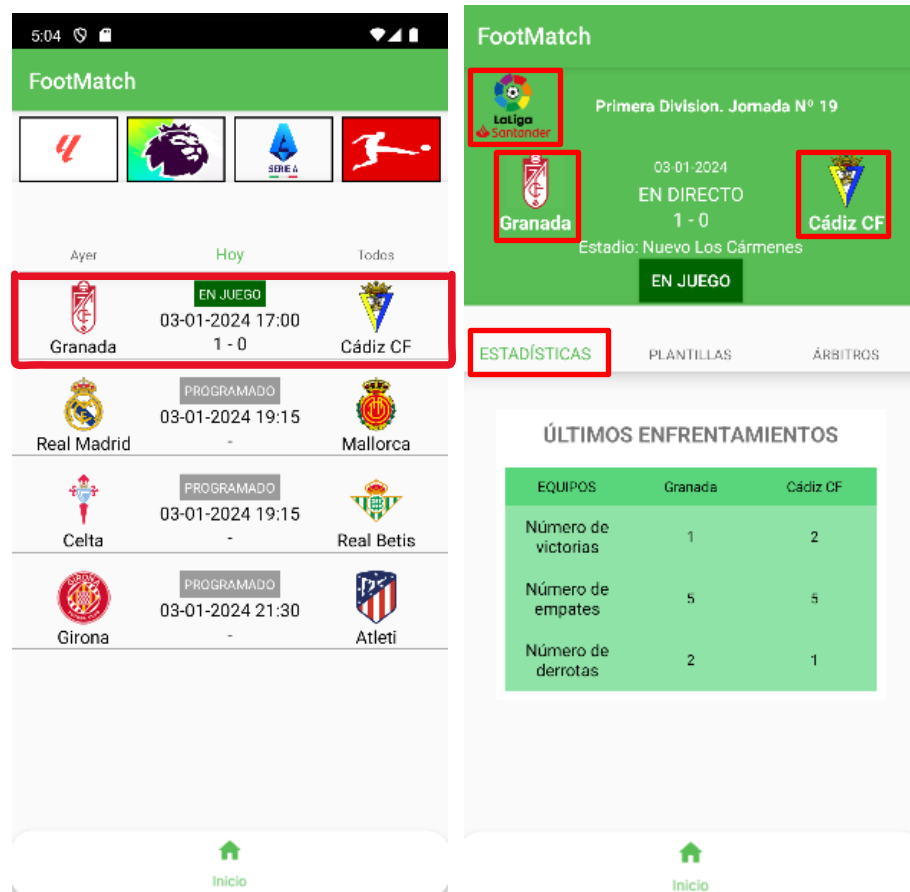


Figura 2. Pantalla (de detalle de un partido) donde se puede apreciar las estadísticas

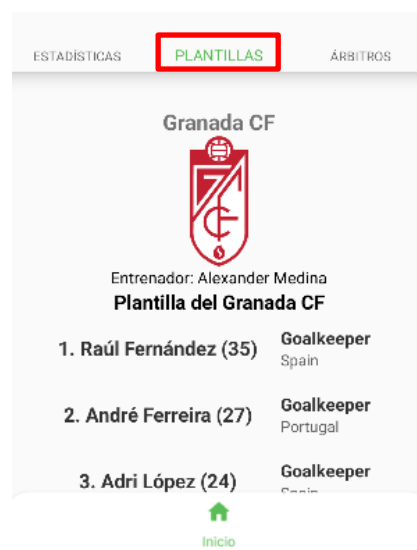


Figura 3. Fragmento de plantillas (en este caso del Granada)

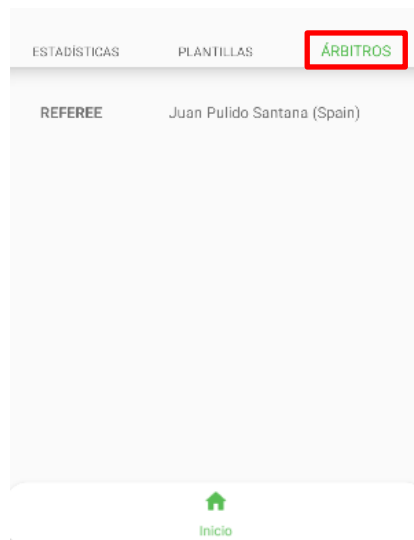


Figura 4. Fragmento de árbitros (en este caso arbitra Juan Pulido, español)

En el caso de que queramos ver más detalles sobre la liga (que es la liga Santander) o los equipos que juegan, debemos de hacer click sobre los iconos de estos y podremos navegar a esta activity. Pero también podemos acceder de la siguiente manera, si seleccionamos inicio volveremos a la pantalla que se mostraba al principio. Vamos a seleccionar la primera liga, se muestra la tabla con la clasificación de todos los equipos y datos de la tabla, y una leyenda con indicaciones del significado de cada columna.

POS	EQUIPO	PG	PE	PP	GF	GC	DG	PT
1		14	3	1	39	11	28	45
2		14	3	1	42	21	21	45
3		12	2	4	36	19	17	38
4		11	5	2	34	21	13	38
5		10	5	3	34	19	15	35
6		8	8	3	30	10	11	32
7		6	10	2	20	19	2	28
8		6	8	5	24	25	-1	26

POS	EQUIPO	PG	PE	PP	GF	GC	DG	PT
1		14	3	1	39	11	28	45
2		14	3	1	42	21	21	45
3		12	2	4	36	19	17	38
4		11	5	2	34	21	13	38
5		10	5	3	34	19	15	35
6		8	8	3	30	10	11	32
7		6	10	2	20	19	2	28
8		6	8	5	24	25	-1	26

POS	EQUIPO	PG	PE	PP	GF	GC	DG	PT
1		14	3	1	39	11	28	45
2		14	3	1	42	21	21	45
3		12	2	4	36	19	17	38
4		11	5	2	34	21	13	38
5		10	5	3	34	19	15	35
6		8	8	3	30	10	11	32
7		6	10	2	20	19	2	28
8		6	8	5	24	25	-1	26

POS	EQUIPO	PG	PE	PP	GF	GC	DG	PT
1		14	3	1	39	11	28	45
2		14	3	1	42	21	21	45
3		12	2	4	36	19	17	38
4		11	5	2	34	21	13	38
5		10	5	3	34	19	15	35
6		8	8	3	30	10	11	32
7		6	10	2	20	19	2	28
8		6	8	5	24	25	-1	26

POS	EQUIPO	PG	PE	PP	GF	GC	DG	PT
1		14	3	1	39	11	28	45
2		14	3	1	42	21	21	45
3		12	2	4	36	19	17	38
4		11	5	2	34	21	13	38
5		10	5	3	34	19	15	35
6		8	8	3	30	10	11	32
7		6	10	2	20	19	2	28
8		6	8	5	24	25	-1	26

POS	EQUIPO	PG	PE	PP	GF	GC	DG	PT
1		14	3	1	39	11	28	45
2		14	3	1	42	21	21	45
3		12	2	4	36	19	17	38
4		11	5	2	34	21	13	38
5		10	5	3	34	19	15	35
6		8	8	3	30	10	11	32
7		6	10	2	20	19	2	28
8		6	8	5	24	25	-1	26

POS	EQUIPO	PG	PE	PP	GF	GC	DG	PT
1		14	3	1	39	11	28	45
2		14	3	1	42	21	21	45
3		12	2	4	36	19	17	38
4		11	5	2	34	21	13	38
5		10	5	3	34	19	15	35
6		8	8	3	30	10	11	32
7		6	10	2	20	19	2	28
8		6	8	5	24	25	-1	26

POS	EQUIPO	PG	PE	PP	GF	GC	DG	PT
1		14	3	1	39	11	28	45
2		14	3	1	42	21	21	45
3		12	2	4	36	19	17	38
4		11	5	2	34	21	13	38
5		10	5	3	34	19	15	35
6		8	8	3	30	10	11	32
7		6	10	2	20	19	2	28
8		6	8	5	24	25	-1	26

POS	EQUIPO	PG	PE	PP	GF	GC	DG	PT
1		14	3	1	39	11	28	45
2		14	3	1	42	21	21	45
3		12	2	4	36	19	17	38
4		11	5	2	34	21	13	38
5		10	5	3	34	19	15	35
6		8	8	3	30	10	11	32
7		6	10	2	20	19	2	28
8		6	8	5	24	25	-1	26

POS	EQUIPO	PG	PE	PP	GF	GC	DG	PT
1		14	3	1	39	11	28	45
2		14	3	1	42	21	21	45
3		12	2	4	36	19	17	38
4		11	5	2	34	21	13	38
5		10	5	3	34	19	15	35
6		8	8	3	30	10	11	32
7		6	10	2	20	19	2	28
8		6	8	5	24	25	-1	26

POS	EQUIPO	PG	PE	PP	GF	GC	DG	PT
1		14	3	1	39	11	28	45
2		14	3	1	42	21	21	45
3		12	2	4	36	19	17	38
4		11	5	2	34	21	13	38
5		10	5	3	34	19	15	35
6		8	8	3	30	10	11	32
7		6	10	2	20	19	2	28
8		6	8	5	24	25	-1	26

POS	EQUIPO	PG	PE	PP	GF	GC	DG	PT
1		14	3	1	39	11	28	45
2		14	3	1	42	21	21	45
3		12	2	4	36	19	17	38
4		11	5	2	34	21	13	38
5		10	5	3	34	19	15	35
6		8	8	3	30	10	11	32
7		6	10	2	20	19	2	28
8		6	8	5	24	25	-1	26

POS	EQUIPO	PG	PE	PP	GF	GC	DG	PT
1		14	3	1	39	11	28	45
2		14	3	1	42	21	21	45
3		12	2	4	36	19	17	38
4		11	5	2	34	21	13	38
5		10	5	3	34	19	15	35
6		8	8	3	30	10	11	32
7		6	10	2	20	19	2	28
8		6	8	5	24	25	-1	26

POS	EQUIPO	PG	PE	PP	GF	GC	DG	PT
1		14	3	1	39	11	28	45
2		14	3	1	42	21	21	45
3		12	2	4	36	19	17	38
4		11	5	2	34	21	13	38
5		10	5	3	34	19	15	35
6		8	8	3	30	10	11	32
7		6	10	2	20	19	2	28
8		6	8	5	24	25	-1	26

POS	EQUIPO	PG	PE	PP	GF	GC	DG	PT
1		14	3	1	39	11	28	45
2		14	3	1	42	21	21	45
3		12	2	4	36	19	17	38
4		11	5	2	34	21	13	38
5		10	5	3	34	19	15	35
6		8	8	3	30	10	11	32
7		6	10	2	20	19	2	28
8		6	8	5	24	25	-1	26

POS	EQUIPO	PG	PE	PP	GF	GC	DG	PT
1		14	3	1	39	11	28	45
2		14	3	1	42	21	21	45
3		12	2	4	36	19	17	38
4		11	5	2	34	21	13	38
5		10	5	3	34	19	15	35
6		8	8	3	30	10	11	32
7		6	10	2	20	19	2	28
8		6	8	5	24	25	-1	26

POS	EQUIPO	PG	PE	PP	GF	GC	DG	PT
1		14	3	1	39	11	28	45
2								

Si seleccionamos el icono del balón vamos a ir a una nueva activity, donde se listarán los rankings de esa liga: los goleadores (con las estadísticas de goles por partido), los máximos asistentes y los máximos goleadores de penalti. Haciendo scroll podemos ver abajo de los goleadores los asistentes, y así sucesivamente hasta llegar a la última sección. Esta pantalla no tiene botón de home, tiene un botón arriba a la izquierda a través del cual podemos ir hacia atrás (a la activity de ligas), aunque también podemos volver arrastrando a la derecha la pantalla.

FootMatch

Primera Division

POS	EQUIPO	PG	PE	PP	GF	GC	DG	PT
1	<div><div></div></div>	14	3	1	35	11	24	45
2	<div><div></div></div>	14	3	1	42	21	21	45
3	<div><div></div></div>	12	2	4	36	19	17	38
4	<div><div></div></div>	11	5	2	34	21	13	38
5	<div><div></div></div>	10	5	3	34	19	15	35
6	<div><div></div></div>	8	8	3	30	19	11	32
7	<div><div></div></div>	5	10	2	20	19	2	28
8	<div><div></div></div>	6	8	5	24	25	-1	26

Champions

Europa

Conference

Descenso

POS

Posición del equipo en la tabla

EQUIPO

Escudo del equipo de la fila

PG

Número de partidos ganados

PE

Número de partidos empatados

PP

Número de partidos perdidos

GF

Goles que el equipo ha marcado

GC

Goles que el equipo ha recibido

DG

Diferencia entre GF y GC

PT

Puntos acumulados por el equipo

Inicio

FootMatch

RANKINGS DE LA LIGA

GOLEADORES

Jude Bellingham

Real Madrid

Goles: 13 (0.72)

18 partidos

Borja Mayoral

Getafe

Goles: 12 (0.63)

19 partidos

Artem Dovbyk

Girona

Goles: 11 (0.65)

17 partidos

Antoine Griezmann

Atleti

Goles: 11 (0.61)

18 partidos

Gerard Moreno

Villarreal

Goles: 9 (0.56)

16 partidos

Álvaro Morata

Atleti

Goles: 9 (0.53)

17 partidos

Robert Lewandowski

Barça

Goles: 8 (0.50)

16 partidos

Iñaki Williams

Athletic

Goles: 8 (0.44)

18 partidos

ASISTENTES

Robert Lewandowski

Barça

Asistencias: 4

16 partidos

Artem Dovbyk

Girona

Asistencias: 4

17 partidos

Iñaki Williams

Athletic

Asistencias: 3

18 partidos

Gerard Moreno

Villarreal

Asistencias: 2

16 partidos

Jude Bellingham

Real Madrid

Asistencias: 2

18 partidos

GOLES DE PENALTI

Borja Mayoral

Getafe

Penaltis: 4

19 partidos

Artem Dovbyk

Girona

Penaltis: 3

17 partidos

Antoine Griezmann

Atleti

Penaltis: 3

18 partidos

Ante Budimir

Osasuna

Penaltis: 3

18 partidos

Gerard Moreno

Villarreal

Penaltis: 2

16 partidos

Figura 6. Pantallas (de rankings de una liga) donde se listan los jugadores más destacados

Por último, en la ventana de clasificación, podemos seleccionar cualquier equipo de la tabla (no aparecen todos los equipos en la pantalla, pero haciendo scroll podemos acceder a cualquiera de estos). Vamos a hacer click sobre el Real Madrid. Destaca la parte inferior donde tenemos un view pager a través del cual haciendo scroll a los laterales podemos ver la lista de jugadores del equipo. También existe un botón para visitar la web del equipo, se nos abrirá en el navegador del dispositivo de manera instantánea.

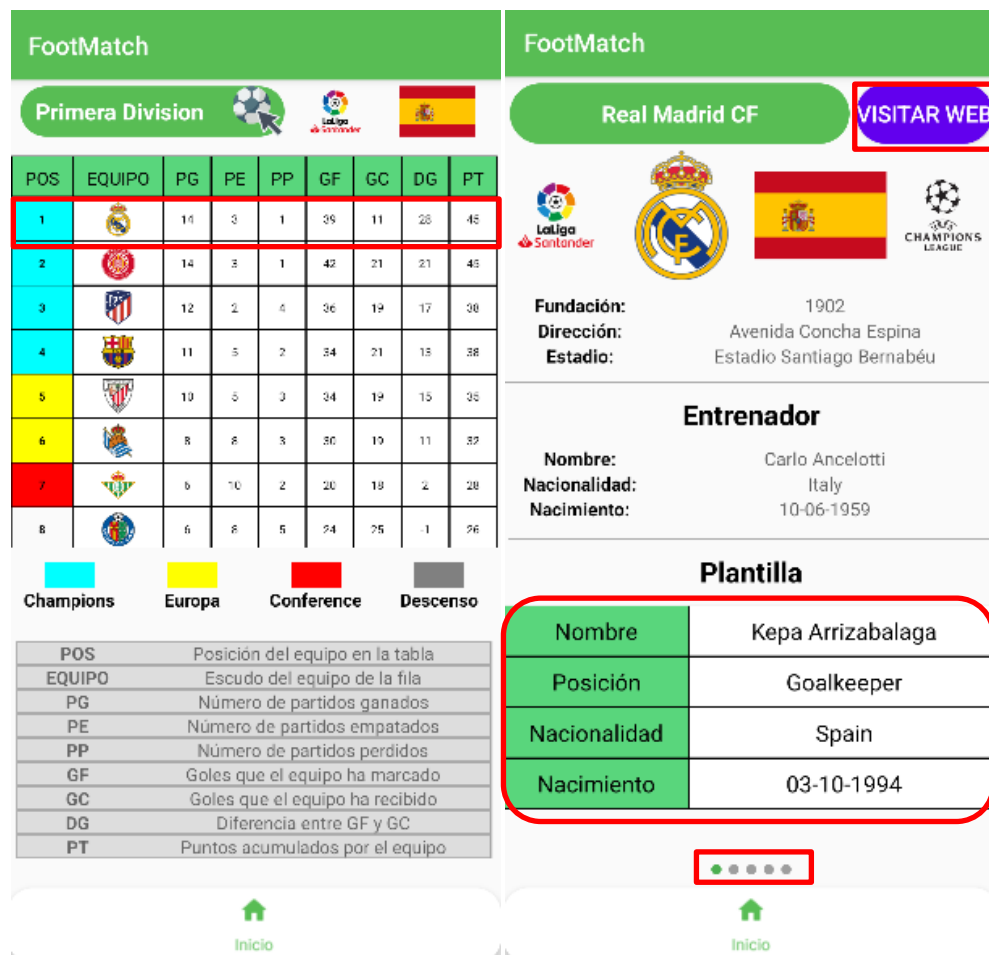


Figura 7. Pantalla (de equipo de una liga) donde se ven los detalles de este

En resumen, el diseño de referencia para las interfaces que forman parte de la aplicación está basado en la interfaz de la aplicación BeSoccer para Android, es decir, el “Look and Feel” (destacando el uso del color verde mayoritariamente).

Dificultades

La mayor dificultad que se nos presentó durante el desarrollo de la aplicación fue el uso gratuito de nuestra API. Esta API está muy limitada, tiene un máximo de 10 peticiones por minuto, lo que nos conllevó una gran dificultad, ya que la información era muy limitada.

Esto nos llevó a otra de las grandes dificultades no solucionadas, como es las fotos de los jugadores. Para poder tener la foto de estos, habría que hacer una llamada por cada jugador, y eso es imposible, por el límite de peticiones comentado previamente.

Decidimos buscar otras API gratuitas. Encontramos 3 opciones, estas eran la versión gratuita de la API de TransferMarket, la API de Wikipedia, y la API de Wikidata.

Con la API de TransferMarket nos encontramos el mismo problema que con nuestra principal, el número de peticiones era muy limitado y no era útil para nosotros.

Con la API de Wikipedia y de Wikidata, encontramos el mismo problema para las dos, si que nos devolvía fotos, y podíamos hacer las peticiones que queríamos, pero no devolvía de forma ordenada, es decir, yo hacía la petición para Jude Bellingham y su foto de perfil era la 5 de la lista, y si hacía esa petición para Borja Mayoral, era la primera, por lo que decimos que no era una opción ya que era algo aleatorio que no podíamos controlar con un algoritmo.

Otra dificultad, fue el trabajo con Kotlin y XML, ya que era nuevo para nosotros, las rutinas al recibir datos de la API y de la base de datos fue un gran quebradero de cabeza, ya que nos costó mucho utilizarlas al principio por ser algo nuevo para nosotros.

Finalmente, otra dificultad fue el cambio de lenguaje de nuestra aplicación, nosotros comenzamos utilizando JAVA, pero según avanzábamos veíamos que se no se nos adaptaba a nuestras necesidades y decidimos cambiar a Kotlin, dentro de este paso el más complicado fue la refactorización de los accesos a la API, y toda su preparación, ya que hubo que rehacerlo desde 0, no fue solamente refactorizar código de un lenguaje para otro.

Deuda técnica

Nuestro gran problema ha sido nuestra limitación del uso de nuestra API, aquí detallo nuestra deuda técnica, con un factor común claro y muy limitante, la capa gratuita de la API:

- **Foto de los jugadores:** Fue una de nuestras intenciones principales en nuestra planificación inicial, pero tras intentar solucionar este problema durante todo el transcurso del proyecto, decidimos que la mejor opción era dejarlo de lado. La única solución factible que vemos para esto sería cambiar la tarifa y nuestra suscripción por una más potente en la API.
- **Equipo favorito del usuario:** En un momento inicial, propusimos que un jugador pudiese marcar un equipo como favorito, y que solo apareciese la información, partidos, noticias sobre su, o sus equipos favoritos. Esto fue descartado ya que la capa gratuita de la API no ofrece partidos, o noticias de un equipo individualmente. La solución para esto sería lo mismo que en el anterior punto, mejorar nuestra suscripción a la API.
- **Login:** En un principio, teníamos pensado crear un Login, pero desechamos esta opción ya que no tenía sentido hacerlo. Habíamos desechado la idea del equipo favorito, por lo que el login sería algo innecesario, ya que por estar registrado o identificado no habría ninguna diferencia. La solución para esto sería implementar un Login en caso de que implementemos el marcado como favorito.
- **Alineación de los partidos:** Durante la planificación del Sprint 2, en la exposición nombramos dar la alineación de los partidos, pero como hasta ahora, la API no nos devolvía esta información, así que después de intentar buscar una solución, decidimos abandonarlo. La solución, mejorar la suscripción de la API.
- **Reducción del número de peticiones en la aplicación:** Hemos mejorado el número de peticiones en nuestra pantalla principal y en los partidos, que es donde más se consume, pero en un futuro podríamos mejorar este límite en la pantalla de ligas y equipos, aunque por el momento no decidimos que esa parte era crítica como si era lo mejorado actualmente.
- **Limitación de la API:** Como podemos ver en los demás casos, el límite de peticiones que podemos hacer por minuto es muy limitado, lo que nos limita mucho el funcionamiento de la aplicación y lo que podemos hacer con ella. Hemos tenido que desechar muchas ideas, y cambiar mucho nuestro planteamiento por esto. La solución para esto es la mejora de nuestra suscripción en la API.
- **Diseño gráfico de la aplicación:** Nuestro diseño de layouts es bastante mejorable, mejorar la sencillez de ellos, y que se vean más claros. Esto podría mejorarse con un refinamiento mejor de ellos.