IES NERVIÓN

NBA MyTeam

Alejandro Gata Barros, Curso 2021-2022

1. Estudio de viabilidad del sistema

1.1. Descripción general del sistema

Se pretende realizar una aplicación cuya temática está asociada al basket, concretamente a la NBA (liga de basket más prestigiosa del mundo), y cuyo único fin es el entretenimiento.

Entre las principales opciones que presenta la aplicación, se encuentra la posibilidad de crear alineaciones (o formaciones de juego) con jugadores NBA (los mejores valorados) para, posteriormente, poder compartir estos resultados con otros usuarios de la app. Estas alineaciones podrán cambiar de sistema de juego y existirán diversas alternativas a la hora de colocar los jugadores en su posición (cada jugador tendrá dos versiones, cada una con una posición distinta).

El sistema de obtención de estos jugadores es a través de sobres limitados, los cuales generarán de forma aleatoria un resultado (jugador) dependiendo de la categoría de dicho sobre (oro, plata, bronce).

Por otro lado, la aplicación también cuenta con un sistema de registro y de login para que los usuarios puedan utilizar la aplicación.

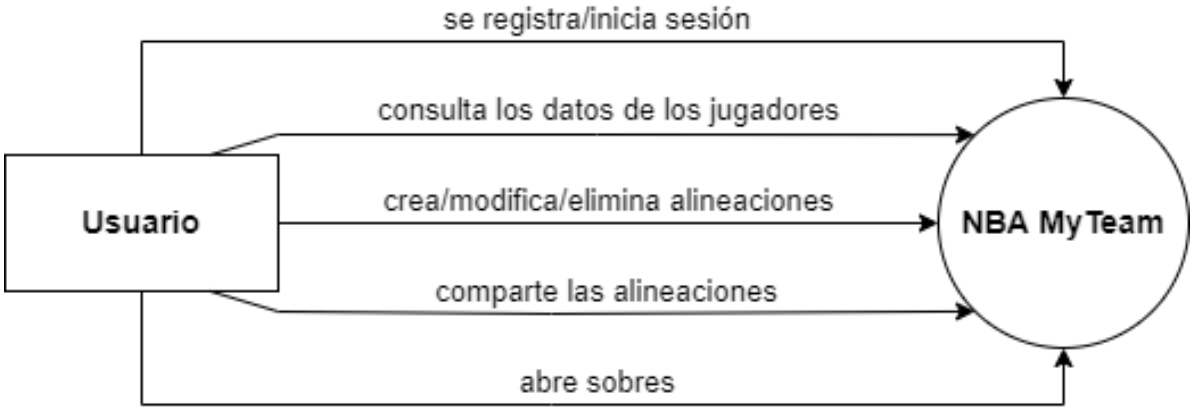
Por último, existen otras opciones como poder consultar los datos de los jugadores (nombre, edad, altura, peso, dorsal, equipo...), cerrar la sesión, etc...

1.1.1. Identificación del alcance del sistema

- **Objetivo**:
    - Desarrollar una aplicación cuyo fin sea el entretenimiento y donde se ayude a conocer el mundo de la NBA y sus jugadores, pudiendo además crear “equipos” con ellos.
- **Requisitos generales:**
    - Debe permitir a los usuarios registrarse, para posteriormente iniciar sesión y así poder hacer uso de la aplicación.
    - Debe permitir cerrar la sesión del usuario cuando ésta se encuentre activa.
    - Debe permitir mostrar los datos principales del usuario activo en ese momento.
    - Debe permitir mostrar una pantalla donde se muestre una breve descripción de la aplicación.
    - Debe permitir ver un listado con los jugadores que se encuentran en el “club” del usuario (jugadores de su propiedad).
    - Debe permitir ver los datos de los jugadores que el usuario tenga en su club.
    - Debe permitir crear alineaciones/formaciones de juego con los jugadores que el usuario tenga en su club.
    - Debe permitir modificar el sistema de las alineaciones.
    - Debe permitir añadir/sustituir/eliminar los jugadores dentro de una alineación propia.
    - Debe permitir consultar los datos de todas las alineaciones.
    - Debe permitir eliminar cualquier alineación que no se encuentre compartida.
    - Debe permitir compartir las alineaciones con otros usuarios registrados.

- Debe permitir ver un listado con todas las alineaciones del usuario, bien sean propiedad de éste o compartidas.
- Debe permitir abrir sobres de distintas categorías para poder obtener así los jugadores.
- Debe permitir ver un temporizador en tiempo real que muestre el tiempo transcurrido desde la última apertura de sobre.

1.1.2. Diagrama de contexto del sistema



1.2. Estudio de la situación actual

1.2.1. Descripción de los sistemas de información existentes

Actualmente existen apps similares para distintas plataformas, pero sobre todo donde más se suelen utilizar es en dispositivos móviles. Además, estas aplicaciones están a la orden del día ya que deportes tan reconocidos como el fútbol o el basket son muy populares entre la gente, y cada vez más usuarios se interesan por aplicaciones de esta temática, donde ver a sus jugadores favoritos y donde poder “ejercer” el papel de manager de un equipo.

Estas apps suelen incluir una gran diversidad de opciones y algunas de ellas son bastantes complejas.

Por ello, lo que se busca con esta aplicación es dar rienda suelta a la imaginación y a la creatividad, gracias a la diversidad de posibilidades que existen a la hora de formar una plantilla con jugadores de distintos equipos y poder “competir” con amigos u otros usuarios. Además, todo ello, acompañado de una interfaz sencilla y atractiva para el usuario.

1.3. Catálogo de requisitos

1.3.1. Requisitos funcionales

Código	Descripción	Prioridad	Fecha
RQF 1	Debe registrar usuarios y validar el acceso (login) a la app	1	19/11/2020
RQF 2	Debe permitir cerrar sesión	1	19/11/2020
RQF 3	Debe permitir crear alineaciones	1	19/11/2020
RQF 4	Debe permitir modificar el sistema de las alineaciones	2	19/11/2020
RQF 5	Debe permitir añadir/sustituir/eliminar jugadores de una alineación	1	19/11/2020
RQF 6	Debe permitir eliminar alineaciones no compartidas	2	19/11/2020
RQF 7	Debe permitir compartir alineaciones con otros usuarios	2	19/11/2020
RQF 8	Debe permitir abrir sobres de distintas categorías	1	19/11/2020
RQF 9	Debe permitir navegar entre las distintas secciones de los menús de navegación	1	19/11/2020

1.3.2. Requisitos de datos

Código	Descripción	Prioridad	Fecha
RQD 1	Deben guardarse los datos referentes al perfil de usuario	1	19/11/2020
RQD 2	Deben guardarse los datos de las alineaciones así como los jugadores incluidos en cada una de ellas	1	19/11/2020
RQD 3	Deben guardarse los datos de todos los jugadores y sus equipos	1	19/11/2020
RQD 4	Deben guardarse los datos sobres las alineaciones compartidas y sus valoraciones	2	19/11/2020
RQD 5	Deben guardarse los datos sobre los jugadores que tiene cada usuario en su club	1	19/11/2020

1.3.3. Requisitos de interfaz

Código	Descripción	Prioridad	Fecha
RQI 1	Debe mostrar una pantalla de registro con un formulario	1	19/11/2020
RQI 2	Debe mostrar una pantalla de login	1	19/11/2020
RQI 3	Debe mostrar un menú principal donde aparezcan las distintas secciones de la app	1	19/11/2020
RQI 4	Debe mostrar los datos principales del usuario activo en ese momento	2	19/11/2020
RQI 5	Debe mostrar una pantalla donde se muestre una breve descripción de la app	3	19/11/2020
RQI 6	Debe mostrar una pantalla donde se muestre un listado de los jugadores que el usuario tenga en su “club”	1	19/11/2020

RQI 7	Debe mostrar una pantalla donde se muestren los datos concretos de cada jugador	2	19/11/2020
RQI 8	Debe mostrar un listado con las alineaciones del usuario (propias o compartidas) junto con un botón que permita crear una nueva alineación	1	19/11/2020
RQI 9	Debe mostrar una “ficha” con los datos de cada alineación junto con dos botones, uno para eliminar, y otro para compartir dicha alineación	1	19/11/2020
RQI 10	Debe mostrar un gráfico que represente el sistema de juego de cada alineación y donde aparezcan las imágenes de cada jugador incluido en ella	1	19/11/2020
RQI 11	Debe mostrar una pantalla con imágenes de los distintos sobres disponibles junto con un botón que permita ver el tiempo transcurrido desde la última apertura	1	19/11/2020

1.3.4. Requisitos no funcionales

Código	Descripción	Prioridad	Fecha
RQNF 1	Deben funcionar en cualquier dispositivo móvil con una versión no primitiva de Android	2	19/11/2020
RQNF 2	Debe funcionar desde cualquier parte siempre que se tenga acceso a la red	2	19/11/2020

1.4. Estudio de alternativas de solución

1.4.1. Descripción de las alternativas de solución

- **Alternativa I**: consiste en desarrollar una aplicación para dispositivos móviles con sistema operativo Android. Los datos se almacenarían en una base de datos SQL y se accedería a éstos a través de una API.
    - *Requerimientos*: se necesita un servidor donde almacenar los datos y donde ubicar la API, pudiendo utilizar los recursos que Azure nos ofrece.
    - *Requisitos*: solo se podrá usar la aplicación en dispositivos móviles con un sistema operativo Android y una versión no primitiva de éste.
    - *Tecnologías*: un motor de BBDD SQL Server, C# para el desarrollo de la API, Java para el desarrollo de la app en Android y XML para el front.
- **Alternativa II**: consiste en desarrollar una aplicación UWP multiplataforma para Windows 10. Los datos también se almacenarían en una base de datos SQL y se accedería a éstos a través de una API.
    - *Requerimientos*: los descritos en el punto anterior.
    - *Requisitos*: la aplicación solo podrá ser desplegada en dispositivos con Windows.
    - *Tecnologías*: un motor de BBDD SQL Server, C# para el desarrollo de la API, C# para el desarrollo de la UWP y front en XAML.

1.5. Valoración de las alternativas

1.5.1. Estudio de los riesgos

- **Alternativa I**

Riesgo	Probabilidad	Importancia
Falta de experiencia de los desarrolladores	10%	1
Incumplimiento de plazos	30%	1
Poca estabilidad	15%	2
Problemas para soportar determinado número de usuarios	15%	3
Escaso aprovechamiento de la tecnología	15%	2
Incompatible con distintos SO	80%	2
Compatibilidad entre dispositivos	30%	2

- **Alternativa II**

Riesgo	Probabilidad	Importancia
Falta de experiencia de los desarrolladores	10%	1
Incumplimiento de plazos	20%	1
Poca estabilidad	15%	2
Problemas para soportar determinado número de usuarios	15%	3
Escaso aprovechamiento de la tecnología	10%	2
Incompatible con distintos SO	80%	2
Compatibilidad entre dispositivos	10%	2

1.6. Selección de la solución

1.6.1. Evaluación de las alternativas y selección

La opción escogida es la primera alternativa (Alternativa I) ya que la mayoría de aplicaciones homólogas que existen actualmente se encuentran desarrolladas en Android y para dispositivos móviles. Además, no se necesitan recursos lejos de alcance y las tecnologías utilizadas son bastante conocidas por el/los desarrollador/es, y por lo tanto, ayuda a cumplir los plazos más fácilmente.

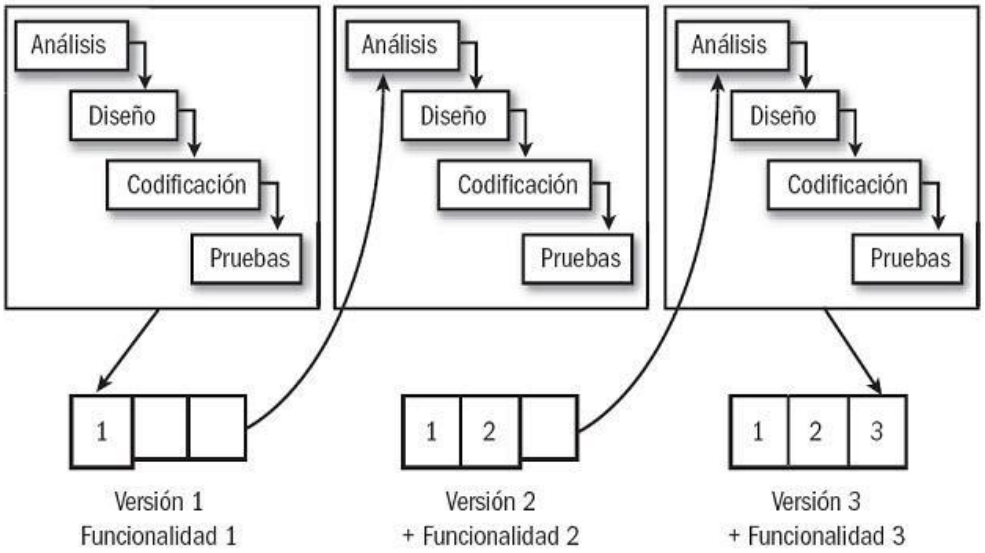
La segunda opción (Alternativa II) no ha sido seleccionada debido a que este tipo de aplicaciones comúnmente no usan un sistema Windows.

2. Gestión del proyecto

2.1. Modelo ciclo de vida

Para el desarrollo del proyecto he utilizado el ciclo de vida incremental. Es decir, he ido añadiendo funcionalidades progresivamente hasta construir el resultado final.

Cada funcionalidad conlleva una serie de pasos para desarrollarla, desde su análisis hasta su diseño, su implementación y la fase de pruebas, hasta lograr un resultado correcto, libre de errores y perfectamente integrado en la aplicación.



2.2. Resumen fases del proyecto

Tarea	Duración aproximada (días)
EVS	5
GPI	75
ASI	50
DSI	75

3. Análisis de sistemas de información

3.1. Descripción general del entorno tecnológico del sistema

Para el desarrollo del sistema se va a crear, en primer lugar, una aplicación Android, implementada en Java y usando XML para las vistas. La estructura general está conformada en varios directorios: activities, adapters, fragments, handlers, repositories, retrofit, utilities y view models.

El mecanismo de comunicación entre las distintas capas es desde las vistas (activities o fragments), donde se origina la petición, pasando por los modelos de vistas (view models) y por los repositorios (repositories), hasta llegar a la capa de retrofit, a partir de donde se envía dicha petición. Seguidamente, al recoger el resultado de esa petición, se realiza el recorrido inverso.

Dicha aplicación ha sido desarrollada en el entorno de desarrollo específico para este tipo de aplicaciones, Android Studio.

En segundo lugar, se ha llevado a cabo el desarrollo de la API, implementada en C#, con framework .NET y usando Visual Studio como entorno de desarrollo.

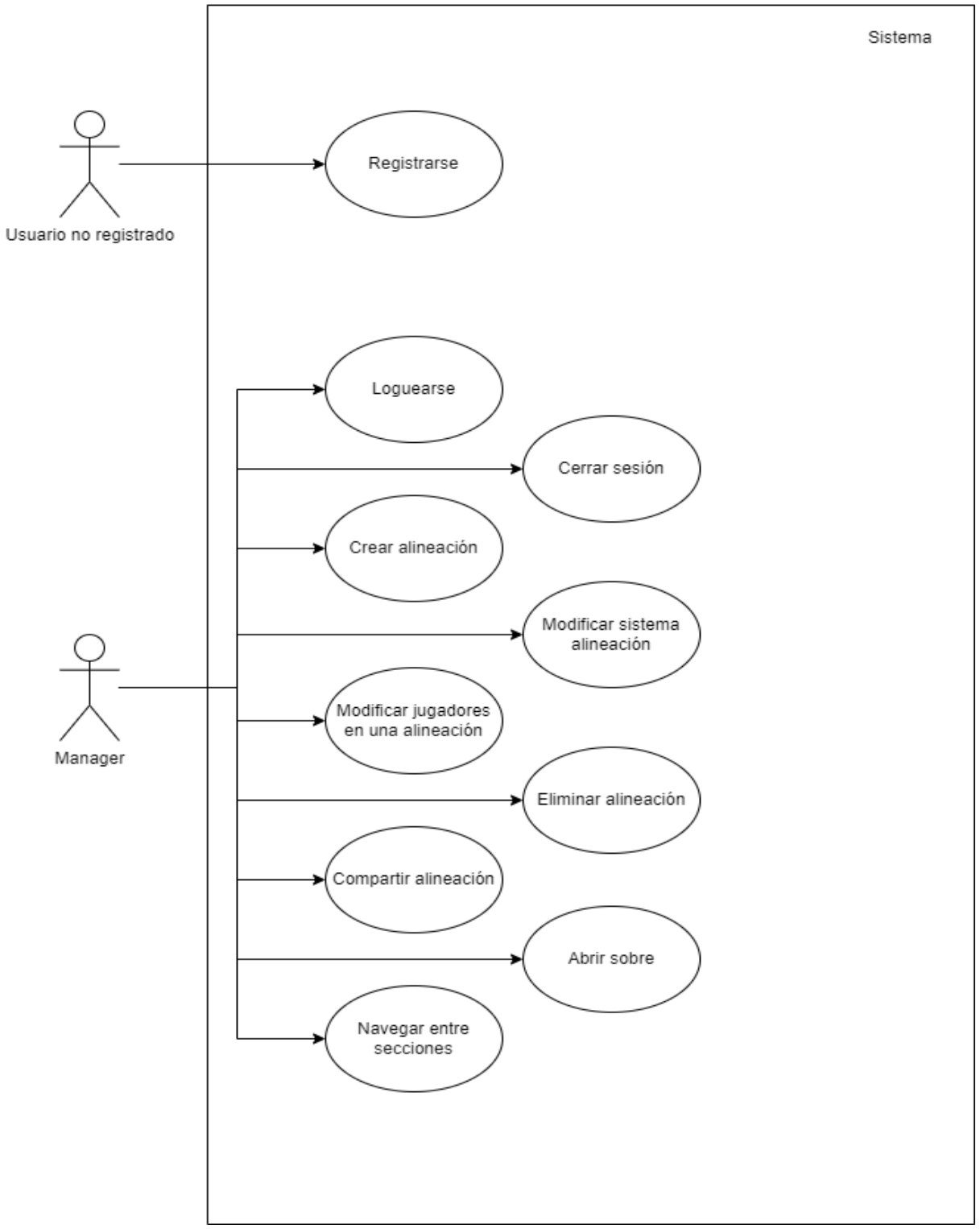
Por último, la base de datos se ha desarrollado en SQL Server y usando Microsoft SQL Server Management Studio.

3.2. Catálogo de usuarios

Código	Nombre	Descripción
Usuario no registrado	Usuario	La única opción a su disposición es registrarse, para poder utilizar todas las funcionalidades de la app
Usuario registrado	“Manager”	Tiene a su disposición todas las funcionalidades de la app

3.3. Casos de uso

DIAGRAMA DE CASOS DE USO



3.4. Especificación de casos de uso significativos

UC1	Registro	
Requisito	RQF 1	
Versión	1.0	
Actor principal	Usuario no registrado	
Personal involucrado	Usuario no registrado	
Descripción	Un usuario anónimo se registra para poder acceder a los servicios de la aplicación	
Precondiciones	Tanto el nick como el correo usado no pueden existir ya en la aplicación	
Postcondiciones	El usuario anónimo queda registrado, y a partir de aquí, puede loguearse y hacer uso de la app	
Secuencia Normal	**Paso**	**Acción**
	1	El sistema ofrece la posibilidad de registrarse
	2	El usuario solicita el registro al sistema pulsando el botón que hace referencia a ello
	3	El sistema muestra un formulario con los datos necesarios para crear un nuevo usuario
	4	El usuario rellena el formulario y pulsa el botón para confirmar
	5	El sistema valida el formulario y registra el nuevo usuario
Extensiones o flujos alternativos	**Paso**	**Acción**
	4	Si el usuario pulsa el botón sin haber rellenado todos los campos obligatorios, el sistema mostrará los errores pertinentes
	5	Si no se cumplen con las restricciones (campos no válidos, nick o correo repetido, contraseña distinta o no aceptar condiciones) no se registrará al usuario y el sistema mostrará los mensajes oportunos
Frecuencia esperada	10/ día	
Temas abiertos	Ninguno	

UC2	Login	
Requisito	RQF 1	
Versión	1.0	
Actor principal	Usuario registrado (“Manager”)	
Personal involucrado	Usuario registrado (“Manager”)	
Descripción	Un usuario registrado se loguea para acceder a la aplicación	
Precondiciones	El usuario debe estar previamente registrado en la aplicación.	
Postcondiciones	El usuario inicia sesión, entra en la aplicación y hace uso de todas sus funcionalidades	
Secuencia Normal	**Paso**	**Acción**
	1	El sistema ofrece la posibilidad de loguearse
	2	El usuario solicita el login al sistema pulsando el botón que hace referencia a ello
	3	El sistema muestra un formulario con las credenciales de inicio de sesión
	4	El usuario rellena el formulario con sus credenciales y pulsa el botón para confirmar
	5	El sistema valida el formulario, verifica las credenciales y permite al usuario entrar en la aplicación
Extensiones o flujos alternativos	**Paso**	**Acción**
	4	Si el usuario pulsa el botón sin haber rellenado todos los campos obligatorios, el sistema mostrará los errores pertinentes
	5	Si los datos introducidos no son válidos o no coinciden las credenciales, el sistema mostrará los mensajes oportunos y no permitirá el acceso a la aplicación
Frecuencia esperada	20/ día	
Temas abiertos	Ninguno	

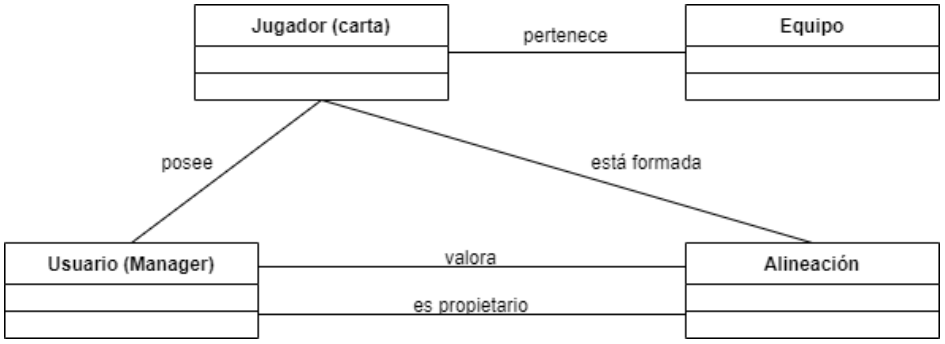
UC3	Crear alineación	
Requisito	RQF 3	
Versión	1.0	
Actor principal	Usuario registrado (“Manager”)	
Personal involucrado	Usuario registrado (“Manager”)	
Descripción	Un usuario activo crea una nueva alineación	
Precondiciones	El usuario debe haberse logueado previamente	
Postcondiciones	El usuario crea la alineación con los datos indicados y se añade a su lista de alineaciones	
Secuencia Normal	**Paso**	**Acción**
	1	El sistema muestra una sección específica para las alineaciones junto con un botón para crear una nueva alineación
	2	El usuario pulsa el botón para crear una alineación
	3	El sistema muestra un pequeño formulario
	4	El usuario rellena el formulario y pulsa el botón para confirmar
	5	El sistema valida los datos introducidos y crea la nueva alineación
Extensiones o flujos alternativos	**Paso**	**Acción**
	4	Si el usuario no rellena los campos obligatorios y pulsa el botón, el sistema mostrará los mensajes de error oportunos y no se creará la alineación
	4	Si el usuario pulsa el botón “Cancelar”, finaliza el proceso y no se creará nada
	5	Si los datos no son válidos, el sistema mostrará los mensajes de error oportunos y no se creará la alineación
Frecuencia esperada	10/ día	
Temas abiertos	Ninguno	

UC4	Modificar jugadores en alineación	
Requisito	RQF 5	
Versión	1.0	
Actor principal	Usuario registrado (“Manager”)	
Personal involucrado	Usuario registrado (“Manager”)	
Descripción	Un usuario activo añade/sustituye/elimina un determinado jugador de una alineación propia	
Precondiciones	El usuario debe haberse logueado previamente y debe existir al menos una alineación a modificar	
Postcondiciones	La alineación queda modificada, ya bien sea porque se ha añadido, sustituido o eliminado un jugador de ella	
Secuencia Normal	**Paso**	**Acción**
	1	El sistema muestra un gráfico que representa los jugadores de cada alineación
	2	El usuario selecciona uno de entre los 5 botones disponibles (cada uno representa un jugador)
	3	El sistema muestra un listado con las posibles alternativas
	4	El usuario selecciona una de ellas
	5	El sistema actualiza la alineación con el nuevo jugador
Extensiones o flujos alternativos	**Paso**	**Acción**
	4	Si el usuario selecciona el botón “Vaciar”, en lugar de una de las opciones mostradas, el jugador se eliminará de la alineación
Frecuencia esperada	10/ día	
Temas abiertos	Ninguno	

UC5	Compartir alineación	
Requisito	RQF 7	
Versión	1.0	
Actor principal	Usuario registrado (“Manager”)	
Personal involucrado	Usuario registrado (“Manager”)	
Descripción	Un usuario activo comparte una alineación con otro usuario registrado en el sistema	
Precondiciones	El usuario debe haberse logueado previamente y debe existir al menos una alineación para compartir	
Postcondiciones	La alineación compartida se añade al listado de alineaciones del usuario “receptor”	
Secuencia Normal	**Paso**	**Acción**
	1	El sistema ofrece la posibilidad de compartir una alineación a través de un botón en la pantalla de detalle de la alineación
	2	El usuario pulsa dicho botón
	3	El sistema muestra un pequeño formulario donde solicita el identificador del usuario al que se desea compartir
	4	El usuario introduce los datos oportunos y pulsa el botón para confirmar
	5	El sistema comprueba que los datos indicados son válidos y existen, y comparte la alineación con el usuario establecido
Extensiones o flujos alternativos	**Paso**	**Acción**
	4	Si el usuario no introduce ningún dato y pulsa dicho botón, el sistema mostrará los mensajes de error correspondientes y no compartirá dicha alineación
	5	Si los datos no son válidos o no existen, el sistema mostrará los mensajes oportunos y no compartirá dicha alineación
Frecuencia esperada	10/ día	
Temas abiertos	Ninguno	

<table>
<tr><td>UC6</td><td colspan="2">Abrir sobre</td></tr>
<tr><td>Requisito</td><td colspan="2">RQF 8</td></tr>
<tr><td>Versión</td><td colspan="2">1.0</td></tr>
<tr><td>Actor principal</td><td colspan="2">Usuario registrado (“Manager”)</td></tr>
<tr><td>Personal involucrado</td><td colspan="2">Usuario registrado (“Manager”)</td></tr>
<tr><td>Descripción</td><td colspan="2">Un usuario activo abre un sobre de una determinada categoría</td></tr>
<tr><td>Precondiciones</td><td colspan="2">El usuario debe haberse logueado previamente</td></tr>
<tr><td>Postcondiciones</td><td colspan="2">El jugador obtenido en dicho sobre se añade al listado de jugadores del usuario</td></tr>
<tr><td rowspan="5">Secuencia Normal</td><td>Paso</td><td>Acción</td></tr>
<tr><td>1</td><td>El sistema muestra una sección con los sobres disponibles</td></tr>
<tr><td>2</td><td>El usuario pulsa sobre uno de ellos</td></tr>
<tr><td>3</td><td>El sistema muestra el resultado con una imagen del jugador obtenido y añade dicho jugador al “club” del usuario junto con el resto de jugadores</td></tr>
<tr><td>4</td><td>El usuario cierra la ventana mostrada</td></tr>
<tr><td rowspan="2">Extensiones o flujos alternativos</td><td>Paso</td><td>Acción</td></tr>
<tr><td>3</td><td>Si no ha transcurrido el tiempo suficiente para abrir un sobre de la categoría seleccionada, el sistema mostrará el mensaje oportuno y no obtendrá ningún resultado</td></tr>
<tr><td>Frecuencia esperada</td><td colspan="2">10/ día</td></tr>
<tr><td>Temas abiertos</td><td colspan="2">Ninguno</td></tr>
</table>

3.5. Modelo de clases



3.6. Interfaces de usuario

3.6.1. Aspectos comunes de la interfaz

La interfaz mostrará aspectos comunes en la pantalla principal (menú principal). Concretamente tendrá, por un lado, un menú “hamburger” donde se mostrarán las opciones de “Cerrar sesión” y “Más información”, para que el usuario en cualquier momento pueda desconectarse y para que pueda ver la información de la app cuando desee.

Por otro lado, tendrá un menú inferior dividido en tres secciones: “Club”, “Alineaciones” y “Sobres”. El usuario podrá navegar entre estas secciones simplemente pulsando el botón del apartado que desee. En la parte superior, se mostrará el nombre de la sección actual y en el resto de la pantalla se mostrará el contenido de dicha sección.

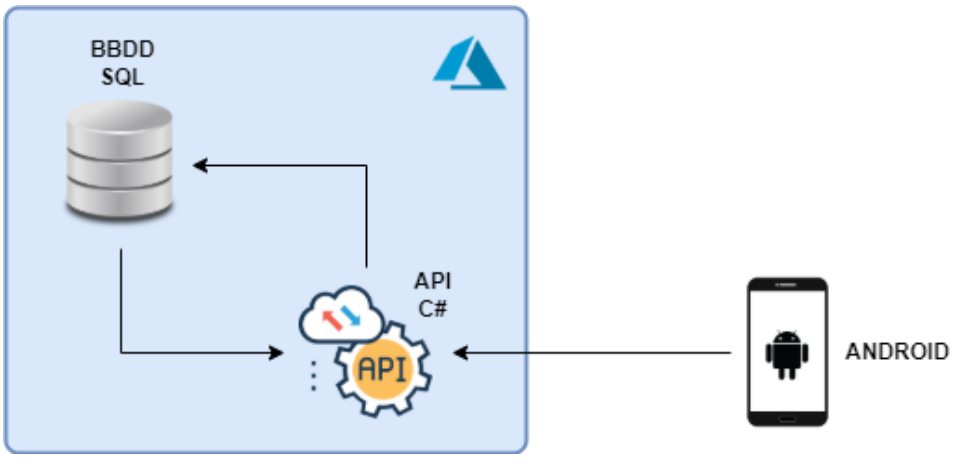
3.6.2. Especificación de pantallas y ventanas

- **Portada**: es la primera pantalla de la app. Muestra una imagen de fondo como portada, el nombre de la app y un botón para entrar.
- **Menú registro y login**: es la siguiente pantalla en caso de que no exista una sesión iniciada. Muestra dos botones principalmente, uno para registrarse y otro para iniciar sesión, junto con un mensaje de bienvenida.
- **Registro**: muestra un formulario con los datos del usuario y un botón para confirmar.
- **Login**: muestra un pequeño formulario con los datos para la identificación (nick o correo electrónico y contraseña) y un botón para confirmar.
- **Información app**: muestra una descripción general sobre la app.
- **Club**: muestra un listado de jugadores (con imágenes) que pertenecen a dicho usuario. En caso de que no existan jugadores, se mostrará un mensaje informativo.
- **Alineaciones**: muestra un listado de alineaciones que pertenecen a dicho usuario. En caso de que no existan alineaciones, se mostrará un mensaje informativo.
- **Sobres**: muestra tres imágenes correspondientes a los distintos sobres disponibles, y un botón para ver el tiempo transcurrido desde la última apertura.
- **Detalle jugador**: muestra los datos del jugador (nombre, edad, dorsal, altura, equipo...)
- **Detalle alineación**: muestra una ficha con los datos de la alineación y un gráfico que representa el sistema de dicha alineación, donde además se muestran los jugadores incluidos en ella en caso de que existan.

4. Diseño de sistemas de información

4.1. Diseño de la arquitectura del sistema

4.1.1. Descripción general del entorno tecnológico del sistema



Los usuarios con un dispositivo móvil Android solicitarán, a través de sus interacciones, diversas respuestas. Para obtener estas respuestas, el sistema de la aplicación enviará peticiones a la API (C#), la cual solicitará los datos que necesite a la base de datos (SQL Server). Esta última, devolverá los datos a la API, y ésta lo hará a la aplicación.

Tanto la BBDD como la API están desplegadas en Azure, y por tanto, se necesita conexión a Internet para utilizar la aplicación.

4.1.2. Catálogo de requisitos de diseño

La aplicación se estructura principalmente en varios directorios:

- Activities, fragments: almacenan las vistas de la aplicación, las cuales observarán los cambios que se produzcan en las propiedades del view model que tengan asociado.
- View models: almacenan los datos que son ofrecidos directamente a la vista. Estos obtienen los datos de los repositorios.
- Repositories: es una capa intermedia entre el view model y la capa de retrofit. Almacenan los datos que posteriormente serán trasladados al view model.
- Retrofit: es la capa encargada de comunicarse con la API. Ésta recoge las respuestas procedentes de la API.

A continuación se detallan los requisitos:

- Requisitos del modelo de datos

Código	Descripción	Fecha
RQMD 1	El modelo donde se guardarán los datos debe responder a un modelo de datos relacional	21/11/2020
RQMD 2	El motor de base de datos utilizado será SQL Server	21/11/2020
RQMD 3	Los datos deben no ser redundantes	21/11/2020
RQMD 4	Ningún usuario del sistema salvo el equipo de desarrollo y un administrador podrá tener permisos de administración sobre el SGBD	21/11/2020
RQMD 5	Se accederán a los datos a través de una API (Azure)	21/11/2020

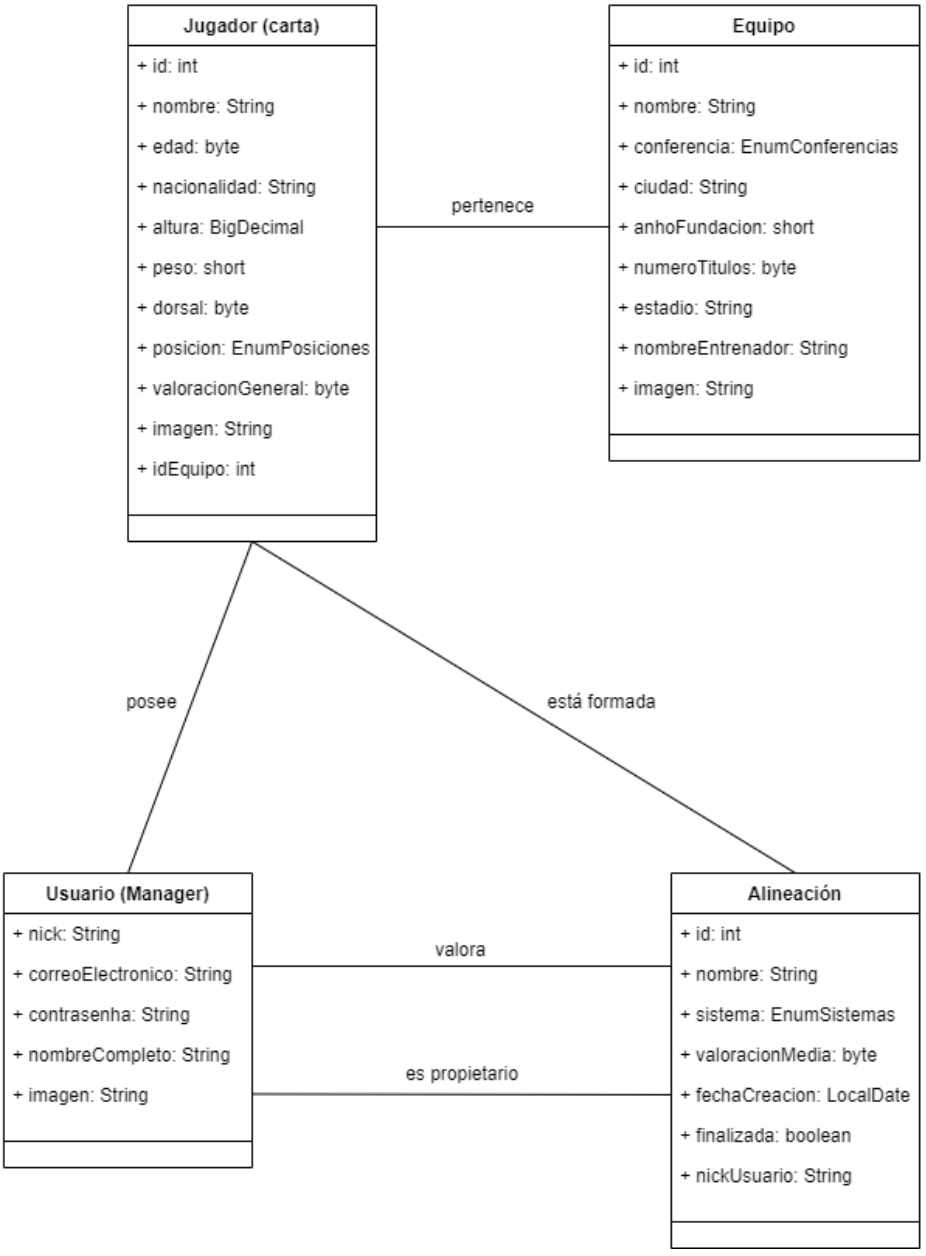
- Requisitos de vistas

Código	Descripción	Fecha
RQV 1	Los cambios de datos a mostrar en las vistas serán controlados mediante observables definidos sobre propiedades del VM	21/11/2020
RQV 2	Los datos de entrada deben ser validados en esta capa para evitar mandar datos erróneos a la API	21/11/2020
RQV 3	La interfaz debe ser intuitiva y sencilla	21/11/2020
RQV 4	La interfaz debe contener información personalizada según el usuario que se encuentre activo en ese momento	21/11/2020

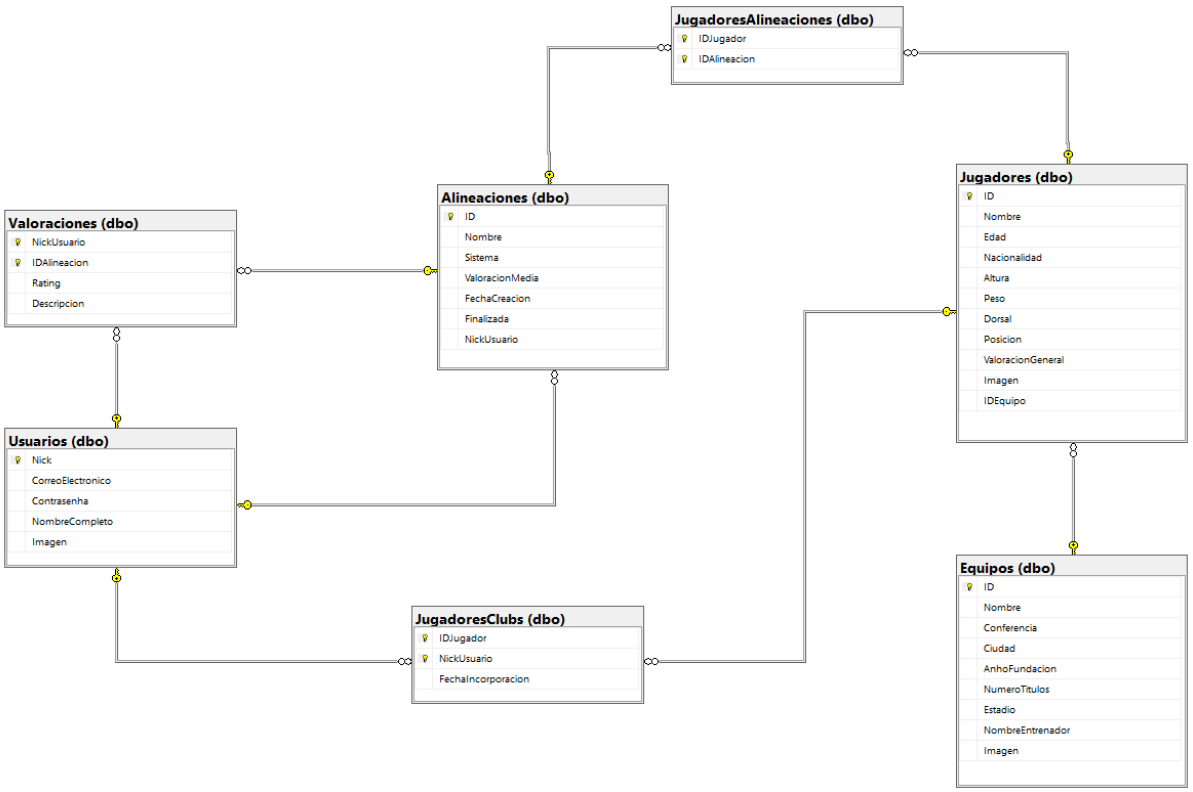
4.2. Modelo de las clases de diseño

A continuación se expone un diagrama que representa las entidades principales del proyecto (no se incluyen las entidades generadas a partir de una relación N:M).

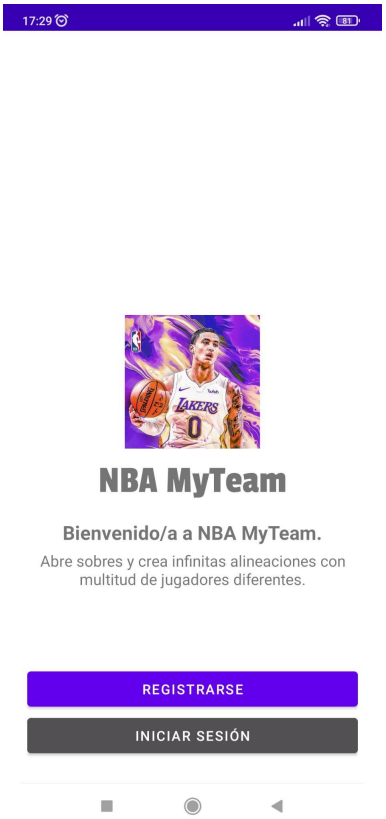
Cada atributo especificado en cada clase contiene sus correspondientes métodos "get" y "set".
No se han colocado en el diagrama por simplicidad y limpieza visual.

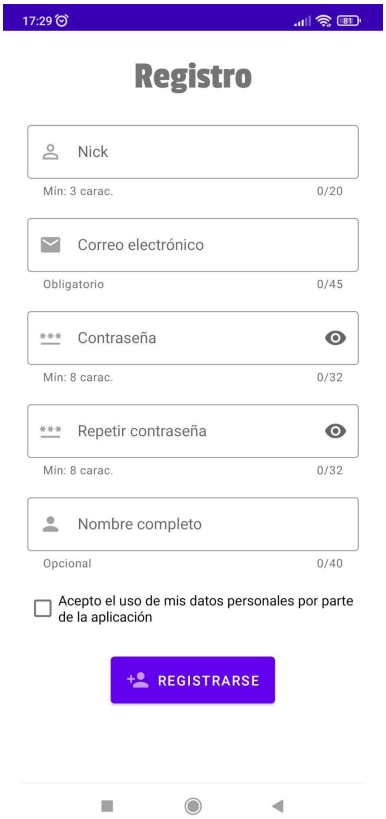


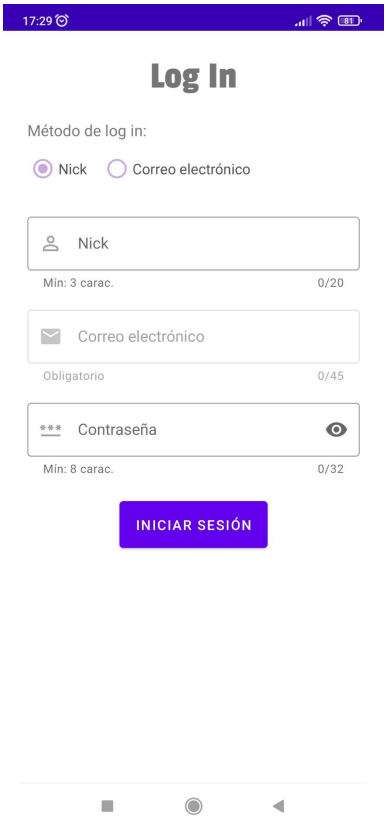
4.3. Modelo físico de datos



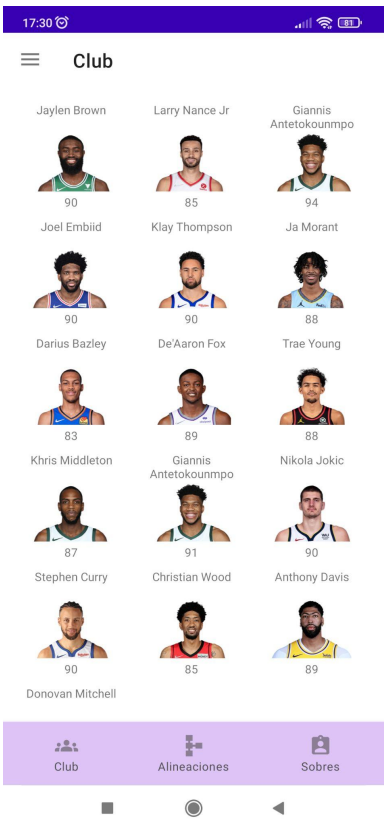
4.4. Diseño de la interfaz de usuario



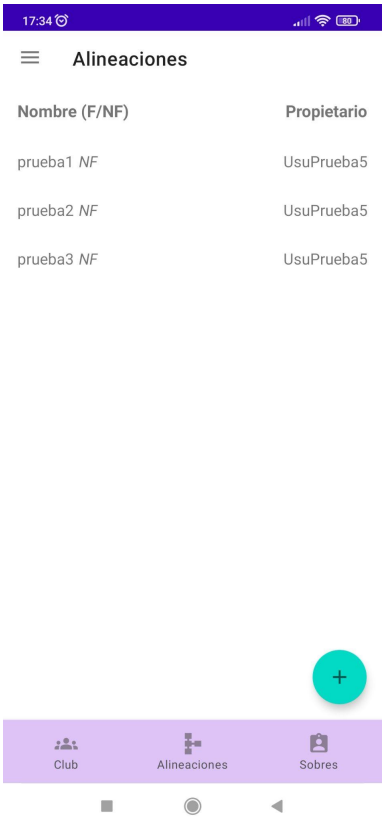


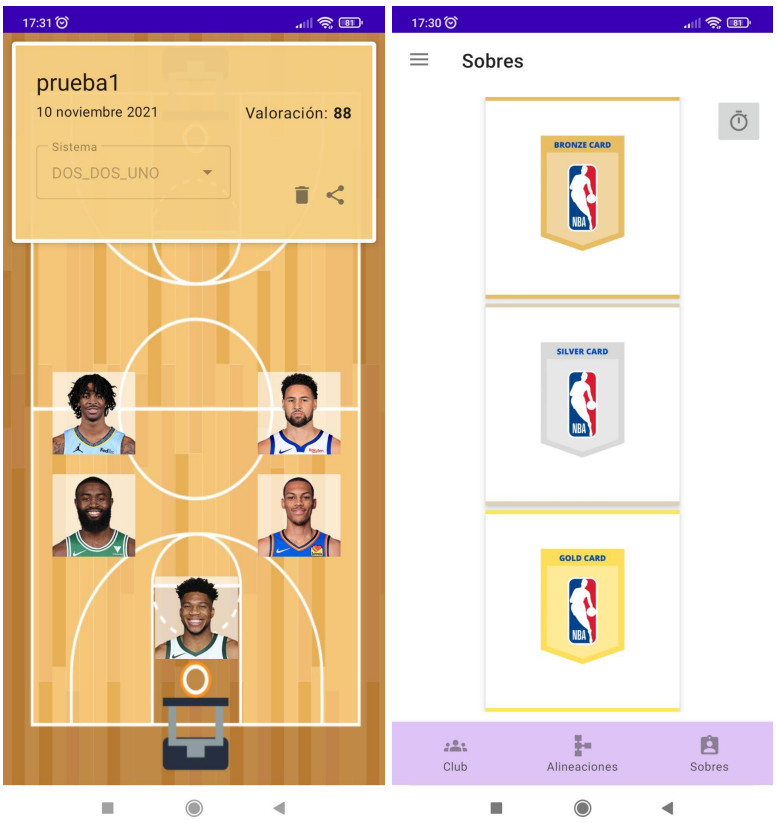



17:30
Información acerca de la app
Se encuentra en una app, NBA MyTeam, cuyo único fin es el entretenimiento y cuya temática está asociada a la mejor liga de basket del mundo, la NBA.
Entre las principales opciones que presenta la aplicación, se encuentra la posibilidad de crear alineaciones con jugadores NBA (entre ellos los mejores jugadores del mundo) para, posteriormente, poder compartir estos resultados con otros usuarios.
El sistema de obtención de jugadores es a través de sobres limitados que generarán jugadores aleatorios dependiendo de su categoría.
Además, también podrá consultar los datos de los jugadores y sus equipos, etc...
NBA


17:30
Club
Jaylen Brown 90
Larry Nance Jr 85
Giannis Antetokounmpo 94
Joel Embiid 90
Klay Thompson 90
Ja Morant 88
Darius Bazley 83
De'Aaron Fox 89
Trae Young 88
Khris Middleton 87
Giannis Antetokounmpo 91
Nikola Jokic 90
Stephen Curry 90
Christian Wood 85
Anthony Davis 89
Donovan Mitchell
Club
Alineaciones
Sobres


17:30
Nombre: Jaylen Brown
Edad: 24
Nacionalidad: Estados Unidos
Altura(m): 1.98
Peso(kg): 101
Dorsal: 7
Posición: ALERO
Valoración: 90
Equipo: Boston Celtics


17:34
Alineaciones
Nombre (F/NF)
Propietario
prueba1 NF
UsuPrueba5
prueba2 NF
UsuPrueba5
prueba3 NF
UsuPrueba5
Club
Alineaciones
Sobres



4.5. Plan de migración y carga inicial de datos

La aplicación no importa datos de ninguna otra aplicación ni de ningún sistema de información existente, por lo que no hay que preocuparse de la migración de datos.

Como carga inicial, se han insertado datos en las tablas de “Jugadores” y “Equipos” para guardar los datos de los equipos y jugadores NBA que estarán disponibles, y en “Usuarios”, con el simple objetivo de tener algunos usuarios de prueba.

Anexo

A continuación, se enumeran algunas de las posibles mejoras que se pueden implementar en el proyecto.

- Añadir la posibilidad de que el usuario pueda cambiar su contraseña y el resto de sus datos de perfil, así como poder eliminar su cuenta si así lo desea.
- Añadir notificaciones al correo electrónico del usuario en caso de que se requiera. Por ejemplo, para cambiar la contraseña.
- Añadir la posibilidad de loguearse a través de una cuenta de Google u otras opciones similares.
- Añadir algunos ajustes, como cambiar el idioma, ajuste de la pantalla...
- Añadir valoraciones a las alineaciones (rating y descripción) para que los usuarios a los que se les comparta una alineación puedan dar una opinión sobre ella.
- Añadir la posibilidad de consultar los datos de los equipos.
- Añadir un filtro donde se pueda realizar una búsqueda de cualquier jugador del sistema.