Universitat Oberta de Catalunya uoc.edu

Máster universitario en Desarrollo de aplicaciones para dispositivos móviles.

f.1 Portada

WinnersPadel

Aplicación para gestionar una liga de pádel desde el móvil y el reloj.

Francisco Javier Nogueras Pardo
Plan de Estudios del Estudiante
Área del trabajo final

Pau Dominkovics Coll
Carles Garrigues Olivella

03/01/2020

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Winners. Aplicación para gestionar una liga de pádel desde el móvil y el reloj*
Nombre del autor:	*Francisco Javier Nogueras Pardo*
Nombre del consultor/a:	*Pau Dominkovics Coll*
Nombre del PRA:	*Carles Garrigues Olivella*
Fecha de entrega (mm/aaaa):	01/2020
Titulación:	*Máster universitario de Desarrollo de aplicaciones para dispositivos móviles*
Área del Trabajo Final:	*M0.659-Trabajo final de máster DADM aula 2*
Idioma del trabajo:	*Castellano*
Palabras clave	*Android* *Gestión* *Pádel*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

Tras varios años gestionando una liga de pádel en la empresa donde trabajo, me encuentro con la necesidad de diseñar una aplicación móvil que me permita llevar un control de los jugadores, los partidos, el calendario y los resultados. Además, como jugador de pádel y poseedor de un reloj inteligente, veo útil la posibilidad de que el reloj me ayude a llevar, en vivo, los puntos del partido.

Combinando ambas ideas tenemos como resultado una aplicación móvil con extensión a los relojes de tipo inteligente que permite gestionar una liga de pádel, tanto a nivel administrador como con otros roles de usuario. El factor diferencial con respecto a otras aplicaciones parecidas es la extensión del reloj que, además de contar los puntos de un partido, permite poderlo arbitrar entre otros jugadores del mismo, a la misma vez que se envía en vivo la información a todo aquel usuario de la aplicación que desee ver en esta el marcador en directo del partido.

Se buscará realizar una aplicación sencilla con las principales áreas de gestión de una liga, y cuya aplicación en el reloj sea una extensión útil de la aplicación móvil. Para ello, ésta tan sólo ayudará a llevar el control de los partidos, así como a mostrar notificaciones que sean de utilidad para el usuario.

Finalmente, se espera que, gracias a la facilidad de uso y a la innovación en los dispositivos de muñeca inteligentes, tenga una buena acogida en el mercado y una buena penetración entre ligas de empresa.

Abstract (in English, 250 words or less):

After several years managing a paddle league in the company where I work, I had the need to design a mobile application that allows me to keep track of the players, the games, the calendar and the results. In addition, as a paddle player and a smart watch holder, I think it would be helpful if the clock could count the match points as I play the game.

As a result of combining both ideas, I want to create a mobile application with an extension to smart-type clocks that will allow everybody to manage a paddle league not only as an administrator, but also as other user roles. This application will offer a new product, which is the extension of the clock. This extension will count the points of a match and, also, the players will referee it. At the same time, that information will be sent to all those users of the application who would like to see the live score of the match.

I will make a simple application with the main areas of management of a league, and whose application on the watch will be a useful extension of the mobile application. This extensión will help to keep track of the matches, as well as showing notifications that are useful to the user.

Finally, thanks to the ease of use and the innovation in smart wrist devices, I hope it will be appreciated in the market and it will have a good entrance in business leagues.

Índice

Lista de figuras

1. Introducción

1.1. Contexto y justificación del Trabajo

Hace seis años comencé con varios amigos del trabajo una liga de pádel. Inicialmente decidimos administrar los datos e información de los partidos con una hoja Excel.

Año tras año los participantes aumentaban, y la organización de esta se hacía más complicada. Por ello empezamos a adoptar alguna aplicación de gestión para poder controlar mejor el torneo. En nuestros requisitos buscamos que fuesen aplicaciones fáciles, de uso gratuito, aunque estuviesen monetizadas con publicidad, y que nos permitiesen controlar de forma eficiente nuestra competición.

Pasamos por diferentes aplicaciones, siendo la más utilizada en esos seis años “Todotorneos.com”. También se valoró algunas de las otras aplicaciones que también existen como “Gestorligas.com”, “competize.com”, “leverade.com”, “ligaprivada.es” o “doleague.com”. Se puede ver una comparativa entre ellas en el apartado “1.7 Estudio de mercado”.

Todas estas aplicaciones, están pensadas como grandes gestores de torneos de todo tipo de deportes. Con un importante grado de complejidad, y muchas opciones genéricas que no terminan de ofrecer la especificidad que para nuestro torneo era necesaria.

Además, como jugador aficionado a este deporte, vi la necesidad de poder llevar la cuenta de los puntos mientras se disputan los partidos, para evitar los comunes y continuos malentendidos.

Estas dos necesidades se juntaron y empezó a nacer en mi cabeza la idea de realizar una aplicación móvil para Android que permitiese a su administrador poder montar una liga de pádel, y a sus usuarios poder conocer calendario, rivales y resultados.

Además, al ser una aplicación específica de este deporte, se podría diseñar una extensión para los dispositivos de muñeca, como los relojes inteligentes, que pudiese ayudar en el conteo de puntos en vivo mientras se disputa el partido.

Este deporte está cobrando auge, y cada día es más practicado [2]. Su sencillez de juego, el entorno cerrado que evita pérdidas de tiempo y el alto nivel social que ofrece a los jugadores, hace que cada día se tenga más en cuenta a la hora de organizar un torneo entre amigos o compañeros.

Por propia experiencia, he podido comprobar como en los 6 años que llevo organizando una liga de pádel en mi trabajo, cada año se ha apuntado más gente, viniendo ésta desde diversos ámbitos, demostrando así que es un deporte abierto a todos los públicos, edades, plural y agradable de practicar.

Es por ello que, tras evaluar mis necesidades a nivel organizativo y deportivo, haber comprobado otras aplicaciones que no me han terminado de convencer y ver la demanda creciente que este deporte está experimentando, he decidido realizar una aplicación que se ajuste a las necesidades de toda mi experiencia

acumulada y específica de este deporte, lo cual considero que será un factor diferencial, ya que como bien dice el refrán “el que mucho abarca, poco aprieta”

Además, incluiré el valor añadido de poder controlar los puntos en vivo mediante los dispositivos de pulsera. En mi caso me centraré en el que actualmente dispongo, un reloj smartwatch con Android Wear 2.0

1.2. Objetivos del Trabajo

1.3. Requerimientos técnicos

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

[1]

Revisando el número mínimo de versión de Android recomendable para efectuar el desarrollo, se puede comprobar que

- Se descarta la versión 10 de Android, lanzada el 3 de septiembre de 2019 y con poca implantación en el mercado, no está ni tan siquiera recogida en los últimos informes de segmentación de Android.
- La versión más usada es la 6.0 seguidas de la 8.1 y 8.0
  - Usar la versión 6.0 implica un mercado del 74.8%
  - Usar la versión 8.0 implica un mercado del 38.7%
  - Descarto la versión 8.1 por motivos técnicos, no dispongo de ningún dispositivo con esta versión.
- La versión de Android escogida para el proyecto será la 8.0 (api 26)

    - El motivo es el de intentar aprovechar las características más innovadoras de Android, a la ver que se llega al mayor número de usuario posible. Android 6.0 se lanzó en 2015. En el momento en que se lance la aplicación, habrán pasado casi 5 años, condicionando a ser anticuada una aplicación cuyo punto fuerte es ser innovadora.

En cuestión al uso de relojes inteligentes, por motivos técnicos para realizar las pruebas, se usará la versión de Android de los relojes que dispongo.

Además, se usará un reloj con pantalla redonda y otro con pantalla cuadrada.

- Huawei Watch 2 2018, pantalla redonda, Wear 2.9.0
    - Lanzado el primer trimestre de 2018
- Lg G Watch, pantalla cuadrada, Wear 1.5
    - Lanzado el 25 de junio de 2014
- La versión escogida para el reloj, será Wear 1.5
    - La versión se justifica para poder acceder al mayor número posible de dispositivos con Android Wear. Los relojes inteligentes suelen tener una vida útil mayor que la de los móviles, al ser aparatos que requieren menos recursos, y cuyas funcionalidades pueden aguantar más en el tiempo. Además, por las características de la app de reloj no se necesitarán de las últimas novedades de la última versión de Wear.

1.4. Requerimientos funcionales

La aplicación desarrollada tendrá que poder gestionar una liga de pádel, para ello se pide que:

- Pueda agregar usuarios, borrarlos y editarlos
- Pueda agregar equipos, borrarlos y editarlos, además de poder incluir y excluir jugadores al equipo
- Pueda generar torneos, agregar y quitar equipos, mostrar un listado de torneos
- Que los torneos que genere sean configurables en diferentes características
- Tenga capacidad para calcular un calendario de partidos, y sus jornadas
- Tenga capacidad para calcular una clasificación de equipos
- Tenga capacidad para recoger información sobre los resultados de los partidos

1.5. Objetivos de la aplicación

La aplicación deberá cumplir los siguientes objetivos para poder ser competitiva en el mercado

- Tener un diseño sencillo y agradable para el usuario
- Ser visualmente bonita y con elementos actuales
- Tener conexión con un reloj Android
- Mostrar notificaciones de usuario
- Que sea configurable por el usuario en determinadas características
- Que esté correctamente testeada para minimizar los fallos
- Que tenga proyección de futuro para poder aumentar sus funcionalidades

1.6. Enfoque y método seguido

Durante el tiempo que me he enfrentado a la problemática que planteo, he ido realizando pequeñas aplicaciones que me permitían resolver algunos de los puntos de mi casuística.

Así pues, realicé una pequeña librería para contar los puntos de un partido de pádel, testeada en varias decenas de partidos. La cual reaprovecharé para este trabajo.

Realicé una pequeña aplicación web que me permitía sortear los partidos de una liga. También testeada en bastantes sorteos. La aplicación web queda fuera del alcance del proyecto, pero reutilizaré la lógica de generación de calendarios si es necesaria.

Por último, realicé una pequeña aplicación que usando la librería anteriormente mencionada, me permitía llevar la cuenta de los puntos de un partido en un móvil Android y en un reloj inteligente.

Estas aplicaciones me servirán de base para poder aplicar los conocimientos adquiridos. Además, me ahorrarán tiempo de desarrollo al tener ya implementados ciertos algoritmos bastante complejos.

La estrategia que seguiré será la de realizar una aplicación Android totalmente nueva, tanto móvil como de reloj. Así podré aplicar mis últimos conocimientos adquiridos en el máster, reaprovechando la lógica de aplicaciones que he realizado anteriormente, como también todo el aprendizaje a nivel de usuario que he obtenido al usarlas.

1.7. Planificación del Trabajo

Las fechas clave de presentación de contenidos son:

- Pec1, propuesta de proyecto: 9 de octubre de 2019

  Se presentará una propuesta analizada, y validada con el tutor

- Pec2, análisis del proyecto: 30 de octubre de 2019
  - Se presentará un estudio de cómo será la aplicación, con la explicación de funcionalidades, casuísticas de la aplicación, diseño de pantallas, diagramas UML, así como una versión alfa de la aplicación que muestre el diseño de esta.
- Pec3, primer prototipo: 11 de diciembre de 2019
  - Se construirán los microservicios necesarios en el servidor, se generarán todas las pantallas de la aplicación móvil y del reloj, se implementará toda la aplicación, y se conectará con el reloj
- Pec4, entrega final del trabajo: 3 de enero de 2020
  - Se pasarán juegos de prueba y se corregirán todos los errores detectados, se terminará la redacción de la memoria, y se realizarán los videos de presentación, así como la entrega definitiva de la aplicación en la tienda de Google.

Teniendo también en cuenta que el tribunal virtual se reunirá del 13 al 17 de enero de 2020.

Para realizar el trabajo en los tiempos y formas que se piden en el proyecto he realizado una estimación de cargas de trabajo, plasmado en un diagrama de Gantt, teniendo en cuenta varios factores como

- Trabajo a jornada completa
  - 8 horas de trabajo que se convierten en 9 de obligación, más 8 de dormir, dejan libres 7 horas
- Compromisos diarios de vida (compras, sacar al perro, limpieza del hogar, cocinar, vida en pareja…)
  - Reduce las 7 horas restantes en 5 en el mejor de los casos.
- Días entre semana
  - Siendo realistas y considerando compromisos inesperados, considero un trabajo de unas 3 o 4 horas al día, teniendo en cuenta que los viernes quizá podría invertir algo más de tiempo.
- Fines de semana
  - En la mayoría de casos he indicado 8 horas de trabajo al día, aunque en algunos casos lo he reducido a 6 ya que soy consciente de que saldrán imprevistos
- Días festivos
  - el pilar de Zaragoza: 11,12,13 de octubre
    - Son las fiestas de mi ciudad, y lo más probable es que tenga poco tiempo disponible para trabajar.
  - cumpleaños de mi pareja: 30 de octubre
    - hay muchas opciones de disfrutar el puente del 1,2,3 de noviembre
  - puente de la constitución: 6, 7 y 8 de diciembre
    - aunque no inicialmente esos días no tengo ningún compromiso, prefiero reservarlos en la planificación
  - navidad: 24, 25, 26
    - aunque pueda sacar algo de tiempo esos días para trabajar, lo más probable es que no lo tenga, y prefiero reservarlos en la planificación
  - noche vieja y año nuevo: 31 de diciembre y 1 de enero
    - apuraré los últimos días de desarrollo, pero soy consciente que será difícil invertir muchas horas esos días

Se puede consultar el diagrama de Gantt en el Anexo 1 donde se indica de forma detallada toda la planificación

El número de tiempo total que se prevé utilizar es de 388 horas.

1.8. Breve sumario de productos obtenidos

Se esperan obtener 5 productos al finalizar el proyecto.

El primero y principal, una aplicación móvil, para el sistema operativo de Android, alojada en la Play Store para que pueda ser usada por todo aquel que la crea útil.

Una aplicación para reloj un inteligente basado en el sistema operativo Android Wear 2.0. Esta aplicación, aunque se destaca como un producto diferente por su carácter especial, en realidad estará integrada dentro de la aplicación para el móvil y se instalará en el reloj al instalar la aplicación principal, si se dispone de un dispositivo que la pueda ejecutar.

Se entregará una memoria pormenorizada con todo el desarrollo de la aplicación, incluyendo explicaciones técnicas y funcionales, así como diagramas UML y explicaciones a bajo nivel para lectores no técnicos.

Por último, se generarán 2 videos, uno promocional de la aplicación que se espera poder incluir en la Play Store para incentivar la instalación de la aplicación. El otro video irá dirigido al tribunal para que puedan evaluar el funcionamiento de la aplicación.

1.9. Estudio de mercado

Como se indica en el punto 1.1 de esta memoria, se analizaron varias aplicaciones de gestión de torneos para usar aquella que mejor se adaptase a nuestras necesidades.

Esta comparación se realizó hace un tiempo, y se eligió la solución que ofrecía “TodoTorneos.com”, por su facilidad de uso, así como por su interfaz, por lo que es la aplicación que más se ha usado, conociéndola bastante a fondo.

Para este proyecto se actualiza la comparación para poder actualizar todo lo que ofrece el mercado.

Para ello se han escogido 5 aplicaciones, a saber:

- TodoTorneos.com
- DoLeague.com
- PadelManager.com
- GestorLigas.com
- Competize.com

Algunas de estas aplicaciones ofrecen planes de pago, teniendo diversas funcionalidades. Es por ello que para homogeneizar la comparativa, en caso de que se ofrezcan estos, se comparará la opción de uso gratuito.

1.9.1. TodoTorneos.com

Características sacadas de la web:

- Completamente gratis y sin necesidad de tener conocimientos técnicos
- Área independiente. Cada torneo tiene su área independiente, o lo que es lo mismo su propia web, tanto para gestionarlo como para publicarlo.
- Diferentes formatos y fondos. Podrá elegir entre diferentes formatos de fondos y colores para la web de su torneo
- Diferentes tipos de competiciones. Podrá crear ligas, rankings, eliminatorias, torneos combinados (clasificación tipo liguilla y Fase Final)
- Publicar toda la información. Se publica la clasificación y resultados por jornada, historial de partidos jugados y pendientes por equipo, página de noticias, normativa, galería de fotos.
- Generación de cruces. Generación automática de cruces de ligas, eliminatorias, sorteo automático de grupos.
- Clasificación. Clasificación personalizada. Personaliza tus puntos por victoria, por derrota, por empate, clasificaciones por partidos ganados, por sets, en base a los resultados la clasificación se calculará automáticamente según tu configuración.
- Utilidades. Galería de fotos, web de contacto, número de visitas.
- Calendario. Establecer y publicar el calendario para los partidos a disputar
- Mapa. Poner un mapa con el callejero de Google maps.
- Gestión de Inscripción. Podrá permitir que los equipos puedan inscribirse a través de la propia web, abrir o cerrar la inscripción a voluntad, establecer cupos de inscripción y gestionar listas de reservas.

<u>Precios</u>

- Sin plan de precios

- No dispone de app móvil

1.9.2. DoLeage.com

- Con DoLeague podrás organizar y gestionar un campeonato profesional (ligas y torneos) de cualquier deporte sin instalación y accediendo desde tu navegador.
- Dispondrás de una página web semi-privada para tu campeonato, con un panel de control para el administrador y un entorno moderno con las

últimas tecnologías para que los participantes puedan acceder a toda la información del torneo en todo momento.

- DoLeague te proporciona además un sistema de inscripción online con notificación automática por email. Podrás seleccionar el método de pago que prefieras, en mano, trasferencia bancaria o incluso usar paypal para cobrar las inscripciones directamente en tu cuenta. Los campeonatos organizados con DoLeague además son completamente configurables, podrás añadir tu propio cartel, tu logo y poner tus colores corporativos.
- Y mucho más, categorías personalizadas, divisiones, grupos, gestión de participantes online, notificaciones y recordatorios automáticos tanto al organizador como a los participantes, calendario de partidos y resultados, cruces automáticos o por ranking y asignación y cálculo automático de las pistas necesarias, etc.
- Pruébalo ahora mismo. Es completamente gratis. Puedes organizar tantos campeonatos como quieras. Revisa nuestros distintos planes de campeonatos para encontrar el que más se adapte a lo que necesitas.

Precios

- Cuenta básica: gratis
- Cuenta pro: 2,49€ / mes
- Cuenta premium: 69€ / mes

- No dispone de app móvil

1.9.3. PadelManager.com

- Espacio exclusivo del club con toda su información e imágenes.
- Difusión de competiciones a toda la comunidad de jugadores a través de notificaciones push.
- Inscripciones online con incidencias horarias de los jugadores.
- Información deportiva detallada de todos los jugadores.
- Estimación de horas de pista en tu competición.
- Creación automática en 5 segundos de los cuadros de competición.
- Horarios de juego adaptados a todos los jugadores.
- Notificación de horarios a todos los jugadores.
- Seguimiento en tiempo real de la competición.
- Red social propia del evento.

- Configuración personalizada de la competición.
- Chats de partidos supervisados por el administrador.
- Resultados autogestionados por los participantes.
- Clasificación automática en tiempo real.
- Búsqueda de pistas disponibles.
- Vinculación con el CRM del club.
- Configuración del método de pago de la reserva.

Precio

- Pago según uso

- Sí tiene app móvil

1.9.4. GestorLigas.com

- Al registrarte en Gestor Ligas te conviertes en un Promotor de eventos deportivos con capacidad para crear, administrar, organizar y gestionar ligas y torneos sin límites con los siguientes servicios gratuitos
- Espacio web propio
- Localización en google maps
- Página de contacto
- Geolocalización
- Crea ligas y torneos
- Puedes crear, administrar, organizar y gestionar ligas y torneos ilimitados, a diferencia de otros gestores deportivos en Gestor Ligas no hay límite de competiciones con la cuenta gratuita
- Gestor Ligas está diseñado para promocionar tu nombre de organizador deportivo, ligas y torneos en los primeros resultados de búsqueda de Google
- En Gestor Ligas puedes añadir y cambiar tu logotipo de administrador deportivo gratuitamente cuantas veces quieras
- En Gestor Ligas disponemos de un sistema propio de Foros que se crean automáticamente para cada Promotor, liga y torneo que podrás administrar para fomentar la participación de tus usuarios
- En Gestor Ligas dispondrás de un espacio web con tus competiciones deportivas que luego puedes personalizar y profesionalizar con la cuenta PREMIUM

- Gestor Ligas pone a tu disposición un formulario de contacto para atender a tus usuarios sin necesidad de mostrar tu email

  Precios

  (consultar vía formulario)

- No dispone de app móvil

1.9.5. Competize.com

- 10 competiciones activas
- Calendario, clasificaciones y cruces
- Alineaciones y estadísticas
- Inscripciones
- Múltiples Campos
- Fotos ilimitadas / Competición
- Fotos ilimitadas / Partido
- Múltiples Admins
- Banners para Patrocinadores
- Importación de ficheros
- Modo TV
- Eventos destacados

Precios

- Amigos: gratis
- Torneos pequeños: 9 € / mes
- Manager profesional: 39€ /mes
- Federaciones y agencias: consultar

- Sí tiene app móvil

1.9.6. Comparativa

Para la comparativa se evaluará el plan gratuito, en caso de tener planes de pago.

	TodoTorneos	DoLeague	PadelManager	GestorLigas	Competize	Winners
Tiene app móvil	✗	✗	✓	✗	✓	✓
Tiene app para reloj	✗	✗	✗	✗	✗	✓
Mensajería entre usuarios	✗	✓	✗	✓	✗	✗
Sistema de notificaciones	✗	✓	✓	✗	✗	✓
Administración de usuarios	✓	✗	✓	✓	✓	✓
Completamente gratis	✓	✗	✗	✗	✗	✓
Multi deporte	✓	✓	✗	✓	✓	✗
Galerías de fotos y videos	✓	✗	✗	✗	✓	✗
Generación de calendarios	✓	✓	✓	✓	✓	✓
Sin limitación de competiciones	✓	✓	✓	✓	✗	✓

f.2.tabla comparativa

1.9.7. Análisis

Las diferentes aplicaciones estudiadas ofrecen servicios bastante parecidos de manera global, aunque con diferentes enfoques, algunos más basados en ofrecer funcionalidades por tipo de contratación de plan, y otras más basados en pago por mayor uso de funcionalidades limitadas.

La principal diferencia entre ellas, como comento, es la monetización. En ellos podemos ver que en algunos casos como con “padelmanger.com” están enfocados directamente a un sector más profesional, con los costes correspondientes y otros como “TodoTorneos.com” están enfocados más hacia el usuario, con una monetización por publicidad.

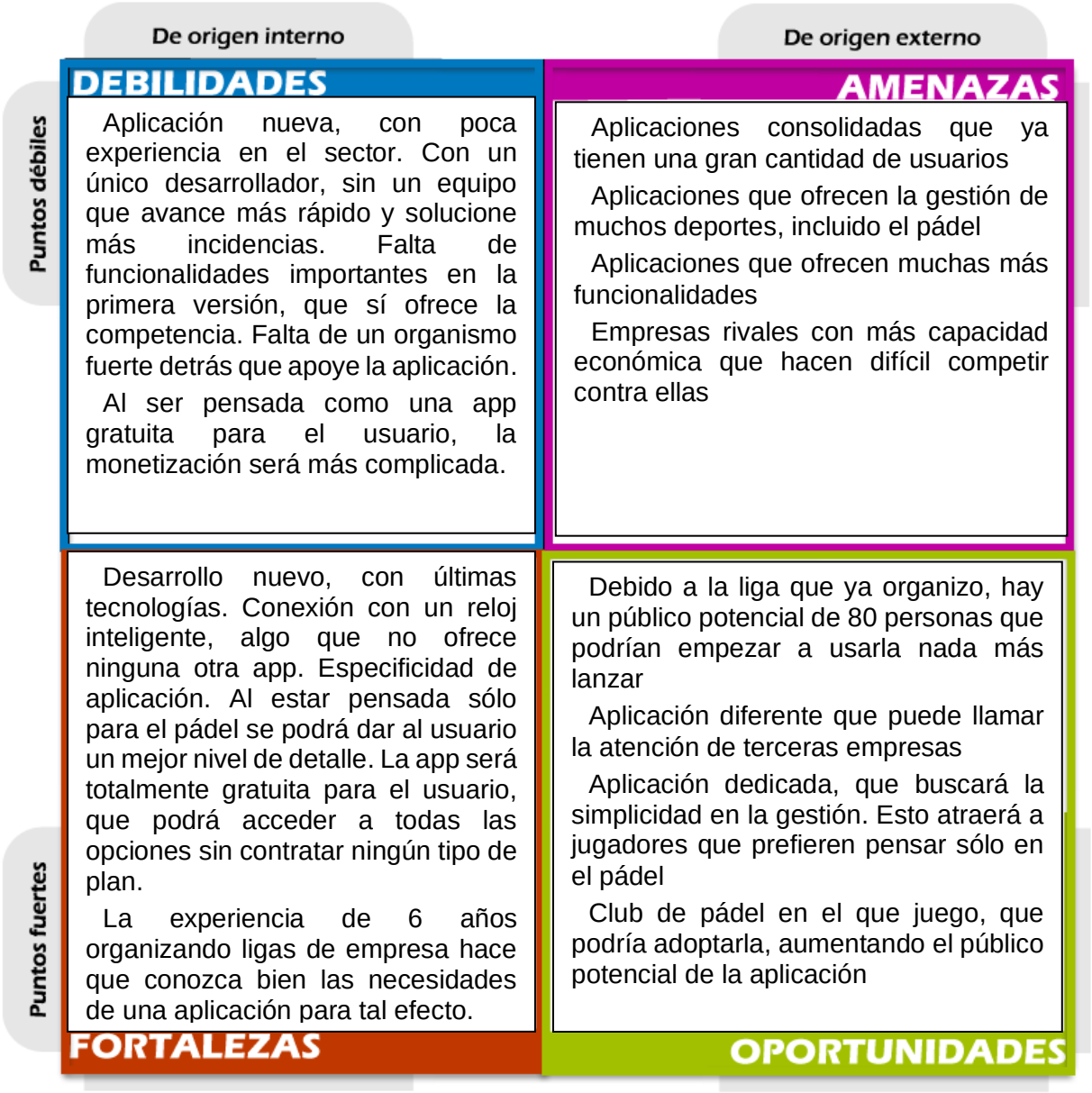
No todas ofrecen aplicación móvil, por lo que pierden competitividad frente a mi proyecto, por no hablar del factor diferencial en mi caso con la aplicación para el reloj inteligente.

Por último, y como se puede apreciar en esta tabla, si comparamos únicamente la parte gratuita de la aplicación, el uso de las aplicaciones queda bastante limitado. Como conclusión, creo que ofrecer una aplicación dedicada al pádel, gratuita para el usuario, y con funcionalidades correctas y optimizadas, puede abrir un nicho de mercado en un sector con bastante competencia.

1.9.8. Dafo

Una vez analizados los productos de la competencia, es conveniente hacer un proceso autocrítico para poder comprender la posición en el mercado que se

ocupará. Para ello a continuación, se muestra una tabla comparativa tipo DAFO, donde se trabajan 4 puntos clave: Debilidades, Amenazas, Fortalezas y Oportunidades



f.3 Dafo

1.10. Breve descripción de los otros capítulos de la memoria

1.10.1. Parte 1, análisis.

La primera de ellas mostrará el enfoque del proyecto, el objetivo del trabajo, la planificación de desarrollo, la comparación de alternativas y el sumario de productos obtenidos.

Además, se plantea una encuesta a usuarios potenciales para poder recopilar información a tener en cuenta a la hora de diseñar la aplicación y sus funcionalidades.

Con todo ello se espera enfocar el producto y planificar su desarrollo para llegar a los objetivos marcados.

1.10.2. Parte 2, diseño de la aplicación.

En la segunda fase, se trabajará el diseño de la aplicación con todo aquello que se considera necesario para poder completar un desarrollo.

Basado en ello, se mostrará un esbozo de las pantallas de la aplicación, se describirán las funcionalidades que se espera que tenga y se pensará el estilo y logos de la aplicación.

Además, se analizarán los datos recogidos de la encuesta de usuarios potenciales.

También se generarán diagramas UML para comprender el funcionamiento de forma más visual.

1.10.3. Parte 3, explicación técnica.

Esta parte del documento tomará un cariz más técnico y en base a esto, se mostrarán diagramas de secuencia y clases de las principales partes destacadas, se explicará el sistema de seguridad elegido, se explicará la librería de conteo de puntos, y se terminarán de completar otra serie de datos técnicos del funcionamiento que sean relevantes.

También se mostrará un Storyboard de los videos, tanto el promocional como el que se presentará ante el tribunal.

Se indicará para finalizar la ubicación de la aplicación en las tiendas correspondientes para que pueda ser instalada.

1.10.4. Parte 4, conclusiones y futuro.

En la última parte de la memoria, se realizará una conclusión de todo el proceso de desarrollo, analizando las partes en las que se han encontrado más problemas.

También se indicarán unas líneas maestras de actuación de cómo se espera seguir evolucionando la aplicación en un futuro.

1.10.5. Parte 5, glosario, bibliografía y anexos.

El documento terminará con la inclusión de todos los datos necesarios de complemento de la memoria, glosario, la bibliografía consultada, y por último todos aquellos anexos que se consideren importantes de adjuntar, como por ejemplo el diagrama de Gantt con la planificación del desarrollo.

2. Diseño

2.1. Estudio de usuarios

A la hora de afrontar el diseño de una aplicación, hay que tener en cuenta a los posibles usuarios futuros de esta y valorar temas tales como: ¿cómo son?, ¿qué tipo de comportamiento tienen en relación a la temática del producto?, ¿qué valoran?, o ¿qué les gustaría encontrar? Entre otras preguntas.

Para ello, he realizado una encuesta dirigida a posibles usuarios de la app, así como a gente que en un principio no estaría interesada en ella, pero que podría dar un enfoque que abriese el diseño a posibles usuarios no valorados inicialmente.

La encuesta está agrupada en las siguientes temáticas:

1. Información del encuestado
    - Para tener los datos de los grupos sociales que se estudian
2. Información sobre el actual dispositivo que tienen
    - Para conocer qué terminales poseen y poder valorar mejor las versiones de la aplicación
3. Información sobre el deporte que realiza, así como de su relación con el pádel
    - Para conocer como de afines podrían llegar a ser con una aplicación relacionada con el mundo del deporte y del pádel
4. Información sobre el tipo de relación que tienen con la tecnología
    - Para conocer qué tipo de uso dan al móvil y las posibilidades que podría tener una aplicación cualquiera de ser exitosa para ellos
5. Información relacionada directamente con la futura aplicación y su uso
    - Para conocer más directamente una respuesta
6. Información sobre pulseras y relojes inteligentes
    - Para conocer qué dispositivos son los que más se usan, así como la relación que tienen con estos

La encuesta ha sido respondida por 33 personas, valorando las respuestas serias y coherentes con las preguntas realizadas.

Esta encuesta se realizó durante 4 días, desde el 8 de octubre de 2019, hasta el 11 de octubre de 2019.

La información que se analizará será de forma genérica basada en las respuestas todos los encuestados. Ésta se analizará sin tener en cuenta sesgos personales, pero valorando las características la muestra usada.

2.1.1. Estudio estadístico

Los datos analizados en este estudio, se pueden consultar en el anexo 7.6

2.1.2. Conclusión

Nos encontramos con un grupo de usuarios potenciales de la aplicación de alrededor de los 30 años, que en su mayoría disponen de teléfonos Android, en su versión 8 o 9.

Tienen una vida deportiva bastante activa y entre los deportes que practican destaca el pádel, el cual realizan en la mayoría de casos a un nivel inicial o medio. Juegan a este deporte por ocio o diversión y reconocen que empezaron a jugar por influencia de terceras personas.

La mitad disputa ligas en la actualidad, lo que podría estar relacionado con que muchos de ellos jueguen más de 10 veces al mes.

La mayoría reconoce no tener interés en contar lo puntos, quizá influidos porque, también la mayoría, admiten que han tenido problemas al llevar el conteo de estos.

En cuestión al uso del móvil, son usuarios selectos, no dependientes de éste, y que no instalan cualquier aplicación. Que si pudiesen preferirían consultar los datos vía web, pero si existiese una buena aplicación para controlar una liga de pádel la instalarían y usarían de buena gana.

Dicen que casi no se conocen aplicaciones para la gestión deportiva y mucho menos del pádel.

Las principales características que buscan en estas aplicaciones son el calendario de partidos para saber cuándo juegan, los resultados que han tenido, la información de los rivales y sus datos de jugador.

Si tuviesen que administrar una liga, dejarían bastante manga ancha a los jugadores a la hora de inscribirse, formar equipos e indicar los resultados, aunque les gustaría poder validarlos.

También destaca que prefieren automatizar tareas como la generación de calendarios.

Por último, los encuestados que disponen de un dispositivo inteligente de muñeca, tienen más relojes que pulseras, por lo que les prima más la interacción con este, que la simple métrica de datos.

En la mayoría de casos son usuarios habituales de estos dispositivos, con más de uno o dos años de uso. Lo que demuestra que, una vez introducidos en el uso habitual, éstas son útiles y se usan de forma regular, sobre todo para controlar la actividad deportiva. Valoran que las aplicaciones del reloj sean claras y concisas sin agobiar con muchos datos.

Con esta información se concluye que un usuario potencial de la aplicación dispondrá de un móvil Android con una versión mínima 8. Y que en una aplicación deportiva para pádel busca que sea clara, que le indique principalmente cuando jugar y los resultados que ha tenido, así como su información propia y la de los rivales, y si no lo tienen consultarán estos datos por otros medios.

Los usuarios de pulseras inteligentes, reflejan con su experiencia que prefieren aplicaciones simples y concretas, y que usan sus dispositivos para controlar su actividad deportiva.

2.2. Escenarios de uso

Juntando el análisis de la información sacada de la encuesta de usuarios, junto con la experiencia acumulada en 6 años de gestión de una liga de pádel en el trabajo, puedo definir los, por ejemplo, siguientes seis escenarios de uso que han incentivado la creación de esta aplicación. Aplicación que resuelve, por ejemplo:

2.2.1. Buscar torneos en los que inscribirse

Nada más terminar la liga de la temporada 2018-2019, Iván decide que este año se volverá a apuntar a la liga de pádel para seguir con su buena progresión. Así que pregunta a Chabi (el organizador de las ligas) cuando empezará la siguiente liga y cuando podrá inscribirse.

Con la aplicación, Iván podrá buscar el torneo de la temporada 2019-2020 y cuando esté publicado inscribirse en él.

2.2.2. Gestionar torneos

Francesc, ha terminado muy motivado la competición y no quiere perder la forma en verano, así que decide montar un torneo de pádel entre los amigos de Borges Blanques, pero no sabe cómo gestionar las inscripciones, calendario, enfrentamientos y resultados de una forma que sea cómoda.

Con la aplicación, Francesc podrá iniciar un torneo propio en el que él será el administrador, y podrá gestionar las inscripciones, así como generar un calendario de partidos, y verificar los resultados de los mismos.

2.2.3. Consultar el calendario, resultados y clasificación de un torneo

Josep tiene un gran nivel de pádel, juega bastantes partidos en un club del que es socio, además participa también en la liga del trabajo. Entre tanto partido a veces se pierde y no recuerda cual es el próximo partido que tiene que disputar, ni cómo han sido los resultados de los anteriores. El club ofrece una aplicación web para ver el calendario, pero no puede consultar los resultados anteriores ni la clasificación, por lo que tiene que esperar a que se publique al final de la fase. Se encontrará con el mismo problema en la liga del trabajo, donde tendrá que enviar un correo al organizador y esperar que este le conteste cuando tenga un rato.

Con la aplicación, Josep podrá organizar su agenda para conocer todos los partidos que juega en todos los torneos, los resultados que ha obtenido y la clasificación en cada instante. Incluso podría solicitar la inclusión del torneo de su club en la aplicación Winners para así poder llevar un mejor control de su participación.

2.2.4. Usar un reloj inteligente para controlar los puntos de un partido

Chabi es un fanático de la tecnología, y dispone de un reloj inteligente que le ayuda a medir, entre otras cosas, su actividad deportiva. Practica pádel de forma habitual, y en la concentración del partido, tras puntos largos a veces no recuerda exactamente el marcador.

Con la aplicación, gracias a la extensión para el reloj, Chabi podrá ir anotando en vivo los puntos que vaya haciendo uno u otro equipo para no perder el marcador.

2.2.5. Aplicación simple, concreta y dedicada

A Alba no le gusta mucho la tecnología ni el deporte, pero por insistencia de su pareja, está valorando si empezar a jugar una liga de pádel ya que es un deporte fácil y divertido. Cuando se entera que tiene que instalar una aplicación en el móvil para poder participar de la liga Alba pone mala cara, no le gusta tener muchas aplicaciones inútiles, sólo algunas específicas, claras y con objetivos definidos.

La aplicación de Winners se ha diseñado pensado sólo en un deporte, concretando cada detalle para no caer en ambigüedades que puedan confundir a los usuarios y con el objetivo de ofrecer soluciones ante los problemas comunes que un jugador de pádel se puede encontrar. Alba encontrará la aplicación útil y le ayudará a disfrutar de este deporte.

2.2.6. Gestión de datos de un partido, fechas y resultados

Anaïs tenía programado un partido para el martes, pero el equipo rival pidió cambiar la fecha porque no les venía bien jugar ese día. Los jugadores de los 2 equipos hablaron y acordaron pedir una nueva fecha y hora para disputar el partido, el jueves de la semana siguiente a las 18.30 y así se lo hicieron saber al administrador para que pidiese reservar la pista al club de pádel. Han pasado los días y Anaïs no sabe o no recuerda si al final se pudo reservar el partido con la nueva fecha o no.

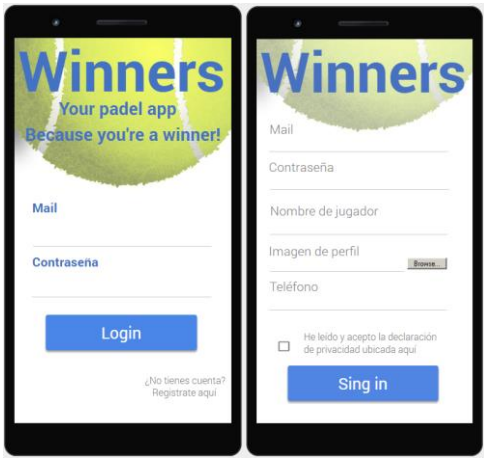
Con la aplicación de Winners, el administrador introducirá la fecha y hora de los partidos, o la modificará si es necesario. También podrá incluir resultados o modificarlos si no son correctos. Así Anaïs tan sólo tendrá que entrar en la aplicación para comprobar si el partido aplazado ya tiene nueva fecha o todavía no.

Estos posibles escenarios de uso, y otros muchos que se podrían producir se resuelven con la aplicación Winners, la cual nace como respuesta a muchas situaciones parecidas a las expuestas y que como organizador de una competición anual me ha tocado vivir, sufrir y resolver de forma bastante habitual.

2.3. Pantallas de la aplicación móvil.

2.3.1. Explicación de funcionalidades

2.3.1.1. Log in / Sing in



El registro se realiza con el email del jugador y la contraseña

Todos los datos son modificables, un usuario tendrá un id interno que será lo único invariable. Se abre la puerta a poder cambiar de email si se cambia de empresa, por ejemplo, o de nombre de jugador si se quiere.

2.3.1.2. Pantalla de inicio

Si no hay datos, no se mostrará nada.

Si se está registrado en un torneo, se verán de este: los partidos programados que se tenga, los resultados y la clasificación de la fase en la que esté activo.

Si se pincha en uno de los partidos programados, la aplicación mostrará la pantalla donde se podrá introducir el resultado.

Si se pincha en uno de los partidos con resultados, se cargará la pantalla con el partido disputado.

Se mostrará una clasificación de la fase actual.

2.3.1.3. Menú lateral

Aparecerá por la izquierda en la mayoría de pantallas. Estará agrupado en 4 bloques

Usuario: edición de los datos personales

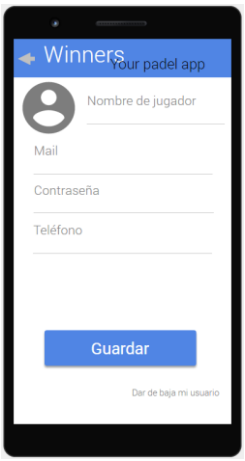
Gestión del torneo: Cambio entre torneos activos, Ver equipos del torneo, Generar nuevo torneo y Buscar un torneo al que inscribirse

Administrador de torneo: Con la modificación de los datos de este, y la validación de notificaciones

Jugar un partido amistoso.

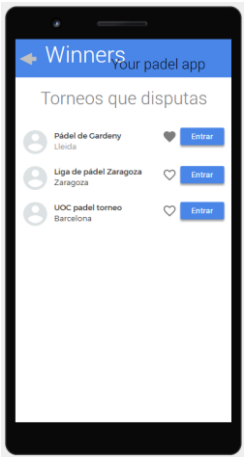
Por último, un botón para cerrar la sesión

2.3.1.4. Modificar los datos de usuario



El usuario podrá modificar todos sus datos, incluido nombre de jugador y correo electrónico, posibilitando así que una persona que se inscribiese con un correo de empresa, si cambia de empresa pueda seguir usando la aplicación con su nueva dirección de correo.

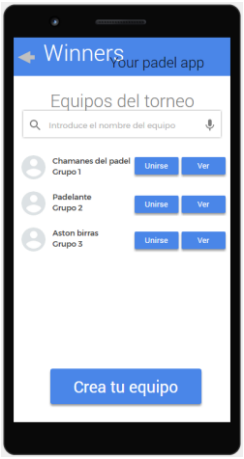
2.3.1.5. Cambiar de torneo



El usuario podrá cambiar los datos de la pantalla principal de la aplicación seleccionado un torneo como principal.

Podría marcar uno, y solo uno, como favorito, haciendo así que, al iniciar la aplicación, siempre sean los datos de ese torneo los que se muestren

2.3.1.6. Equipos del torneo



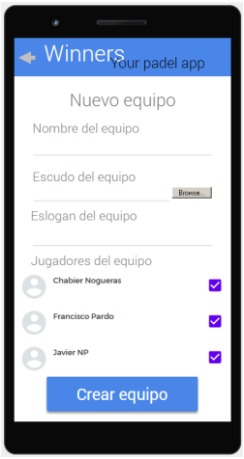
El usuario podrá ver todos los equipos inscritos a un torneo, y podrá solicitar la incorporación a uno de ellos.

La incorporación deberá ser aprobada por el administrador de la competición.

Podrá también ver los jugadores de un equipo.

Y si lo desea, podrá crear su propio equipo

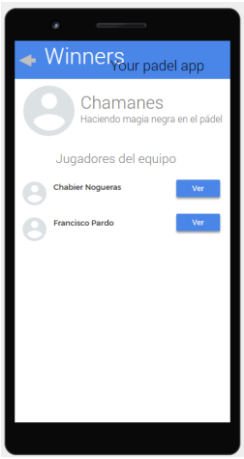
2.3.1.7. Crear equipo



Para crear el equipo habrá que poner un nombre, un escudo si se quiere y un eslogan.

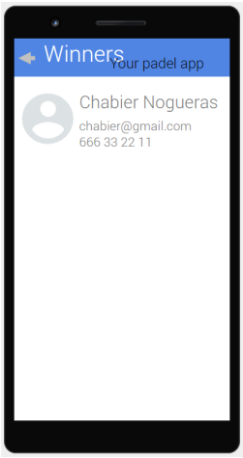
Se podrán elegir los jugadores que formarán parte del equipo de forma inicial, de entre todos los jugadores inscritos en el torneo, aunque cualquier jugador podría solicitar también la incorporación a un equipo.

2.3.1.8. Ver equipo



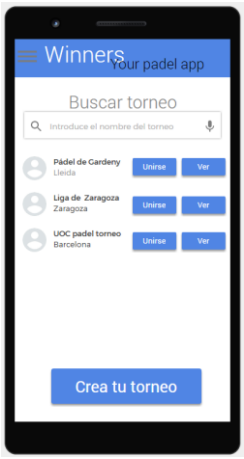
Se podrá visualizar un equipo, con su logo, nombre y eslogan, así como los componentes que lo componen.
Podría incluir más información como los partidos jugados, resultados o estadísticas.

2.3.1.9. Ver jugador



Mostrará la información pública del jugador, como es la imagen de perfil, el nombre, correo y teléfono.
Podría incluir más información como los partidos jugados, resultados o estadísticas.

2.3.1.10. Buscar torneo



Mediante un buscador el usuario podrá indagar entre los diferentes torneos que existen en la aplicación para encontrar el que desee, ya sea para acceder a él y ver el estado, o para unirse y participar en él.
Si no encuentra ninguno que se ajuste a sus necesidades, el usuario podrá crear su propio torneo.

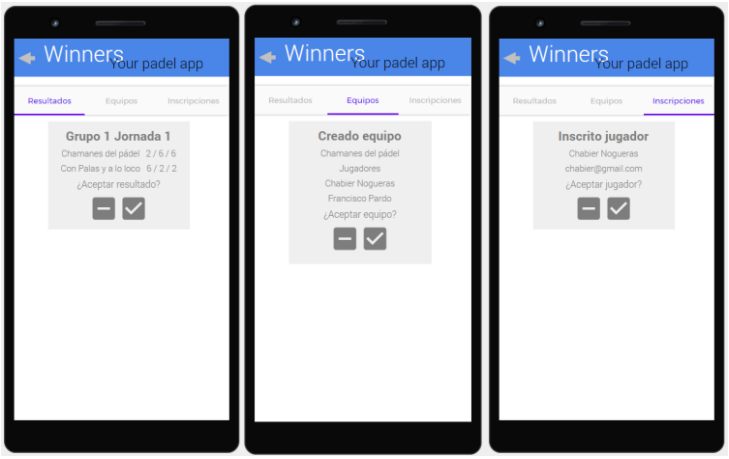
2.3.1.11. Generar nuevo torneo

En caso de que el usuario decida generar su propio torneo deberá rellenar una serie de datos, como son el nombre del torneo, una descripción o eslogan, la fecha de inicio prevista, así como la de conclusión.
También deberá indicar 3 opciones, si el torneo es público o solo es accesible por invitación, si la inscripción está abierta o cerrada, además de si el torneo ha finalizado, aunque se supone que está no será marcada.
Por último, deberá incluir las fases de las que constará el torneo.

2.3.1.12. Agregar fase de grupos / play-off a un torneo

Al agregar las fases, el usuario deberá introducir los grupos de equipos que creará.
Cada grupo tendrá un número de equipos que el administrador deberá incluir.
Se podrán crear tantos grupos como se vea necesario.

2.3.1.13. Validaciones de Resultados, equipos e inscripciones



El administrador del sistema es el responsable de validar las inscripciones de usuarios al torneo, el registro de los equipos que jugarán, y de dar el visto bueno a todos los resultados de los partidos.
La misión de este, es la de proteger el sistema de posibles errores humanos.

2.3.1.14. Introducir resultado de un partido

Cuando un usuario dispute un partido, deberá guarda el resultado en la aplicación. Para ello dispondrá de esta pantalla donde podrá indicar el resultado de cada set.
Tendrá la opción de disputar el partido. Al activar esta funcionalidad, si dispone de un reloj inteligente, este se activará automáticamente para dejarle introducir los puntos desde el móvil, o desde el reloj

2.3.1.15. Controlar un partido

A la hora de controlar un partido, el usuario deberá indicar quién ha realizado el punto, presionando encima del número de puntos actuales del equipo.
Así el contador irá avanzando conforme se vayan realizando los puntos, almacenando internamente toda la información relativa a tiempos, y discurrir del partido.

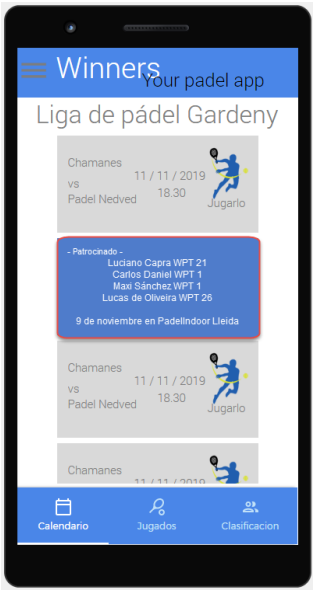
2.4. Monetización

A la hora de diseñar una aplicación, hay que tener en cuenta los gastos que esta puede tener y los ingresos para compensar esos gastos, así como para poder sacar un rédito económico. Es por ello que es importante definir una estrategia de monetización, la cual, en nuestro caso, se ha decidido que tenga las siguientes características

- totalmente gratis para el usuario
- financiación con publicidad y patrocinios
- publicidad no invasiva integrada en la aplicación
- prohibidos banners a pantalla completa o con tiempos de espera
- filtrando la publicidad a público objetivo de ciudades, regiones o países

Con estas premisas se han pensado los siguientes puntos:

2.4.1. Pantalla inicial



- Pestaña de calendario

Un club de pádel podría hacer publicidad de un partido importante, informado de asistentes y fecha, además de estar enlazado a su página web con más información.

Un club de pádel podría ofrecer un partido a un precio especial para los 4 jugadores que se apunten.

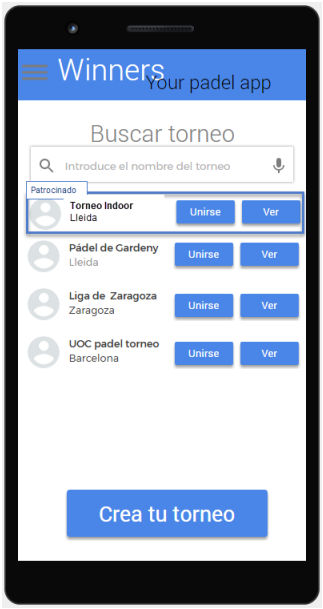
- Pestaña de partidos jugados

Una empresa podría introducir un banner entre los resultados, con el mismo formato que en el calendario, para promocionar sus productos

- Pestaña de clasificación

Una empresa podría introducir un banner al principio o final de la clasificación para promocionar sus productos

2.4.1.1. Buscar torneos



Un club podría promocionar uno de sus torneos apareciendo como destacado entre las búsquedas

Una empresa podría generar todo un torneo y patrocinarlo entero, estableciendo su marca comercial como nombre de competición, con nombres de equipo de sus productos, fases con nombres comerciales de ellos, incluso regalando camisetas y el precio de las pistas.

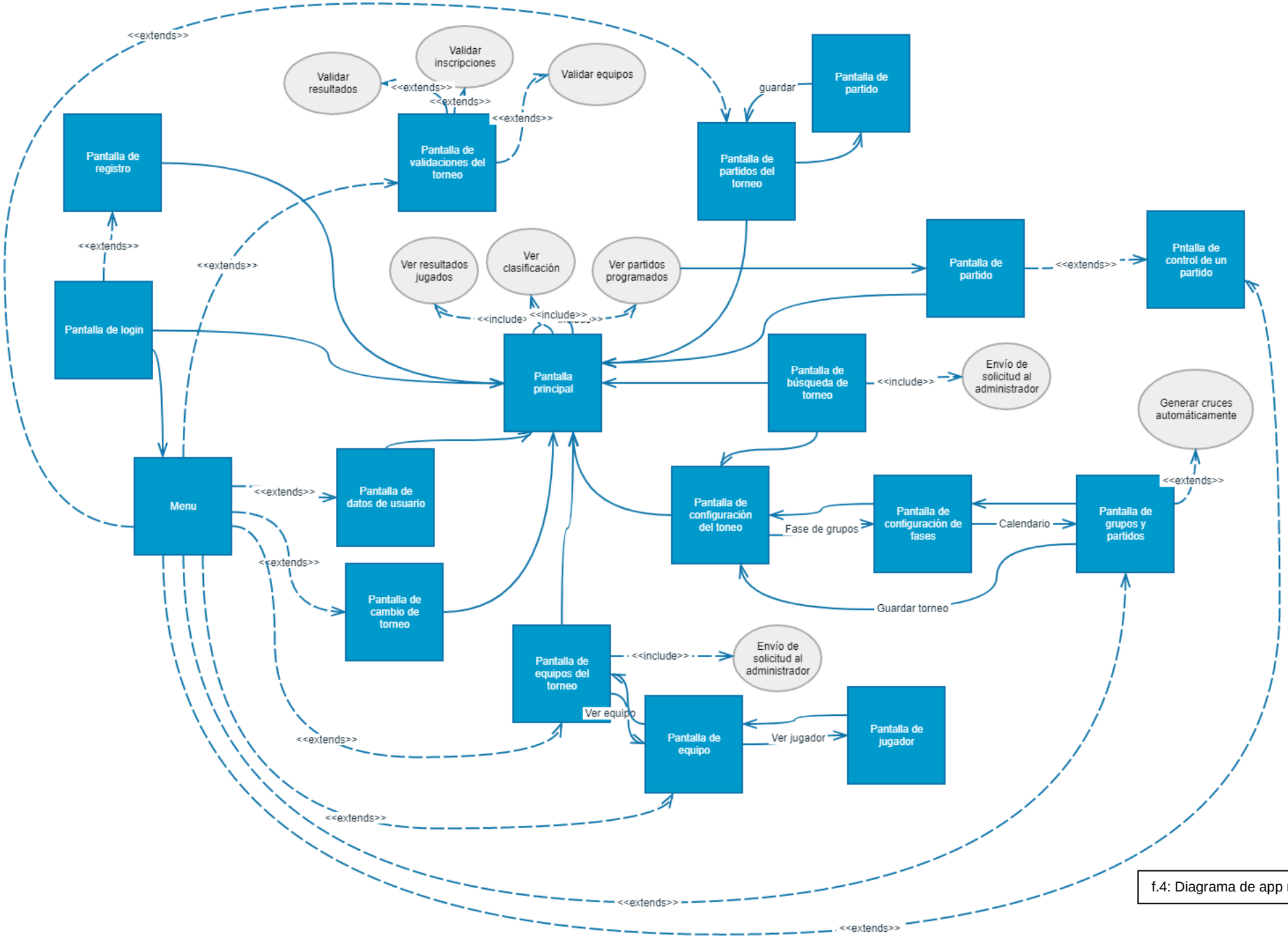
En este caso, también podrían salir destacados en las búsquedas.

2.4.2. Partidos

Única y exclusivamente a la hora de seguir un partido en directo, una empresa podría incluir banners publicitarios en la parte inferior de la pantalla, en la ubicación del botón “Finalizar partido” que sólo vería el usuario que estuviese arbitrando el partido.

2.5. Diseño de la aplicación del móvil

En el siguiente diagrama de pantallas podemos seguir la navegación entre pantallas que está definida en la aplicación.



f.4: Diagrama de app móvil

2.6. Pantallas de la aplicación del reloj

2.6.1. Explicación de funcionalidades

La aplicación del reloj será simple y totalmente orientada a la gestión de resultados, como demandan los usuarios del estudio que tienen experiencia de uso con estos dispositivos.

Arranque:

Cuando la aplicación arranque, se verá una pantalla de bienvenida con el logo de la app.

Inicio:

A continuación, se mostrará la pantalla inicial, que será siempre la que te da la opción de controlar un partido amistoso. Recordemos que la principal función de la parte del reloj es controlar resultados de partidos, así que, lo principal es poder acceder rápido a controlar un partido, sea del tipo que sea.

Arrastrando con el dedo a la pantalla de la derecha, se mostrará un listado de los próximos partidos programados de cualquiera de los torneos que el usuario juegue, para poder acceder rápidamente al partido que te tocaría. Se podrán consultar también los últimos resultados.

Arrastrando de nuevo a la derecha, entraremos en la configuración inicial del partido donde nos encontraremos 2 opciones

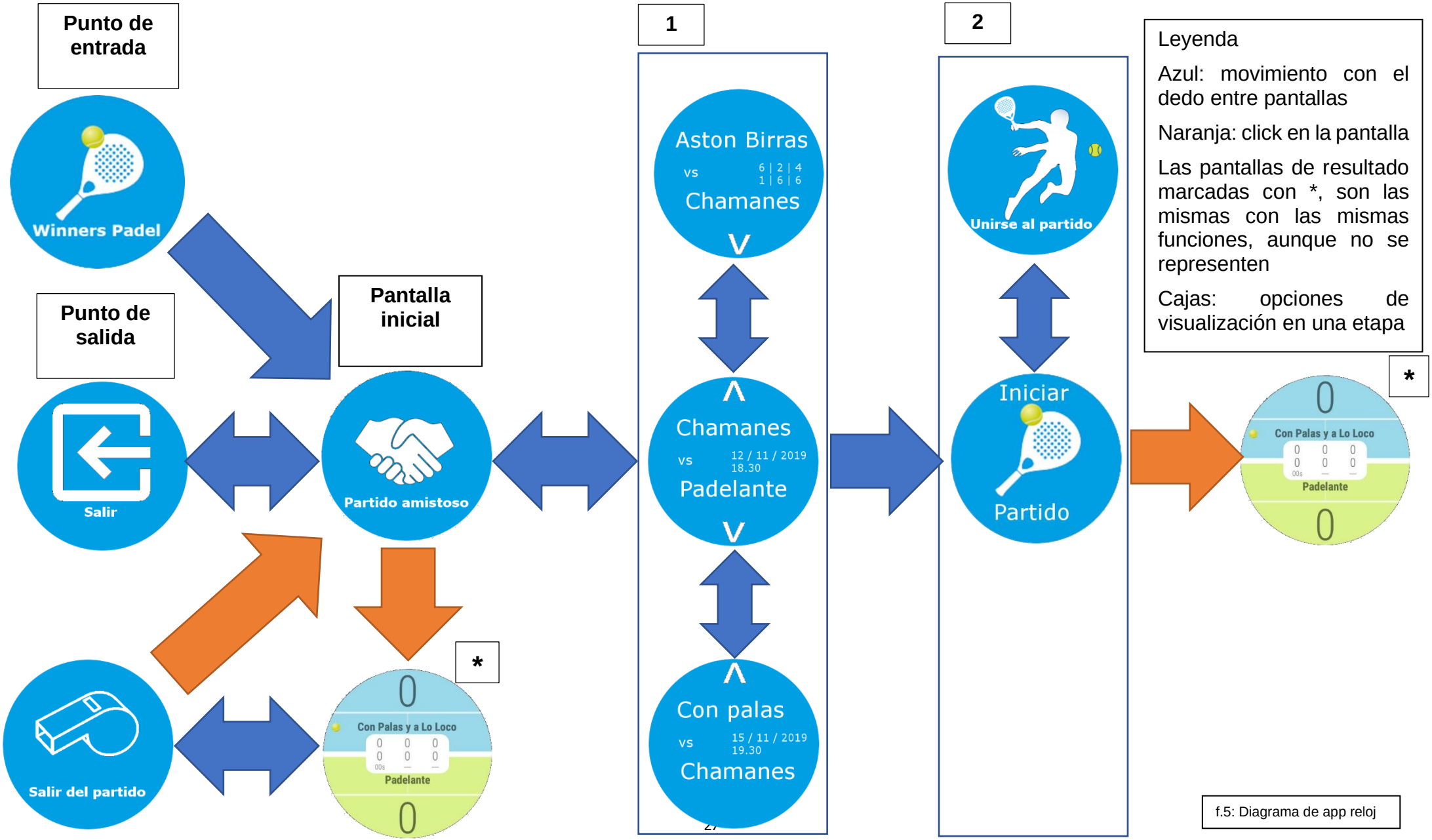
- Inicial partido: El usuario inicia el partido para controlar el marcador.
- En caso de que otra persona se una al partido, este usuario inicial será el único que podrá introducir los puntos.
- Unirse al partido: Si un usuario ha iniciado un partido, y tiene conexión a internet, y otro usuario que se una al partido, también con conexión a internet, podrá ver en su dispositivo el marcador, pero no podrá controlar los puntos.

Si el usuario arrastra el dedo hacia la izquierda, se le mostrará una pantalla de finalización del partido. Al hacer click dará el resultado como válido y llevará al usuario de nuevo a la pantalla de inicio.

De la pantalla de inicio, arrastrando a la izquierda, aparecerá la opción de salir de la app.

2.6.2. Diseño de la aplicación del reloj

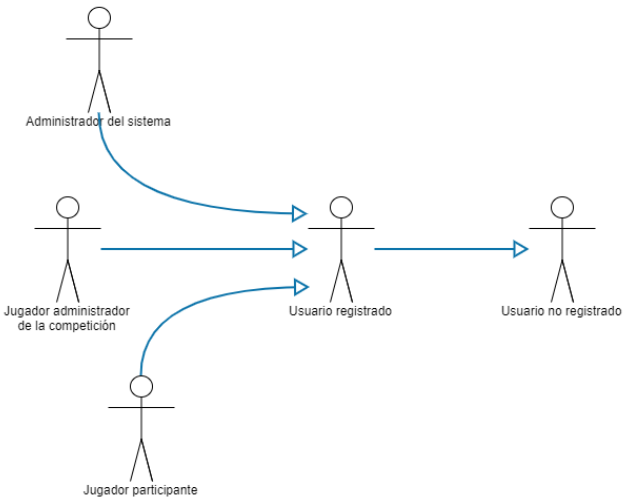
En la siguiente hoja podremos apreciar la organización de pantallas, así como el desplazamiento propuesto entre estas.



f.5: Diagrama de app reloj

2.7. Diagramas UML

2.7.1. Diagrama de actores



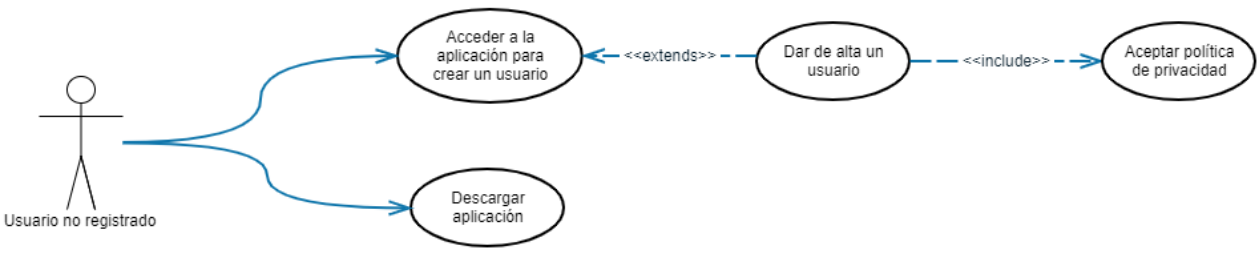
f6 Diagrama de actores

Actor	Características del usuario
	Funcionalidades
Usuario no registrado	Usuario ajeno a la aplicación, que tendrá que crearse una cuenta para poder acceder
	- Descargarse la aplicación - Acceder a ella para crearse un usuario
Usuario registrado	Usuario con cuenta en la aplicación, que podrá participar de torneos, así como generarlos.
	- Iniciar sesión - Cerrar sesión - Editar sus datos de usuario - Buscar torneos en los que participar - Cambiar entre los torneos del usuario - Generar torneo - Inscribirse en un torneo - Inscribirse en un equipo - Jugar partidos y controlarlos - Ver próximos partidos programados - Ver últimos resultados - Ver la clasificación de la liga

Usuario administrador del sistema	Usuario que administrará la aplicación, modificando si es necesario las bases de datos de usuarios y competiciones
	- Acceso a la Base de Datos. - Modificación de torneos, equipos y usuarios. - Corrección de incidencias
Administrador de la competición	Usuario que adquirirá este rol al haber generado una competición, o cuando otro administrador de competición le asigne el rol. Podrá aceptar jugadores, partidos y resultados
	- Agregar Fases de grupos - Agregar Fases de play-off - Publicar torneo - Cerrar el plazo de inscripción - Finalizar torneo - Modificar el nombre y eslogan de la competición, así como la fecha de inicio y fin - Publicar / despublicar torneo - Validar resultados - Validar inscripciones - Validar equipos
Jugador participante	Usuario principal de la aplicación, que jugará partiros y anotará resultados. Podrá ver sus próximos partidos, sus resultados anteriores, y podrá así mismo, cambiar entre los torneos en los que participe para informase de la actividad de ellos
	- Introducir resultados de partido - Utilizar la app del reloj

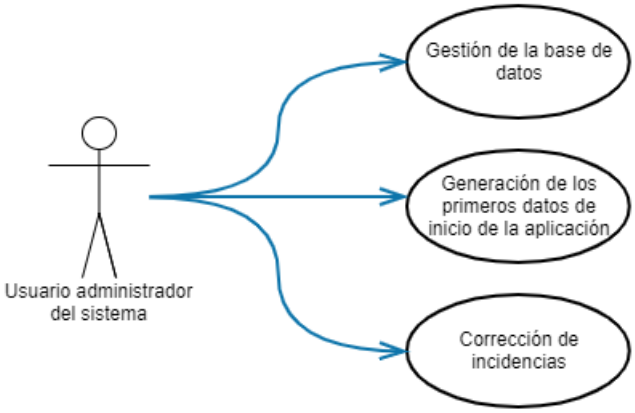
2.7.2. Casos de uso

2.7.2.1. Usuario no registrado



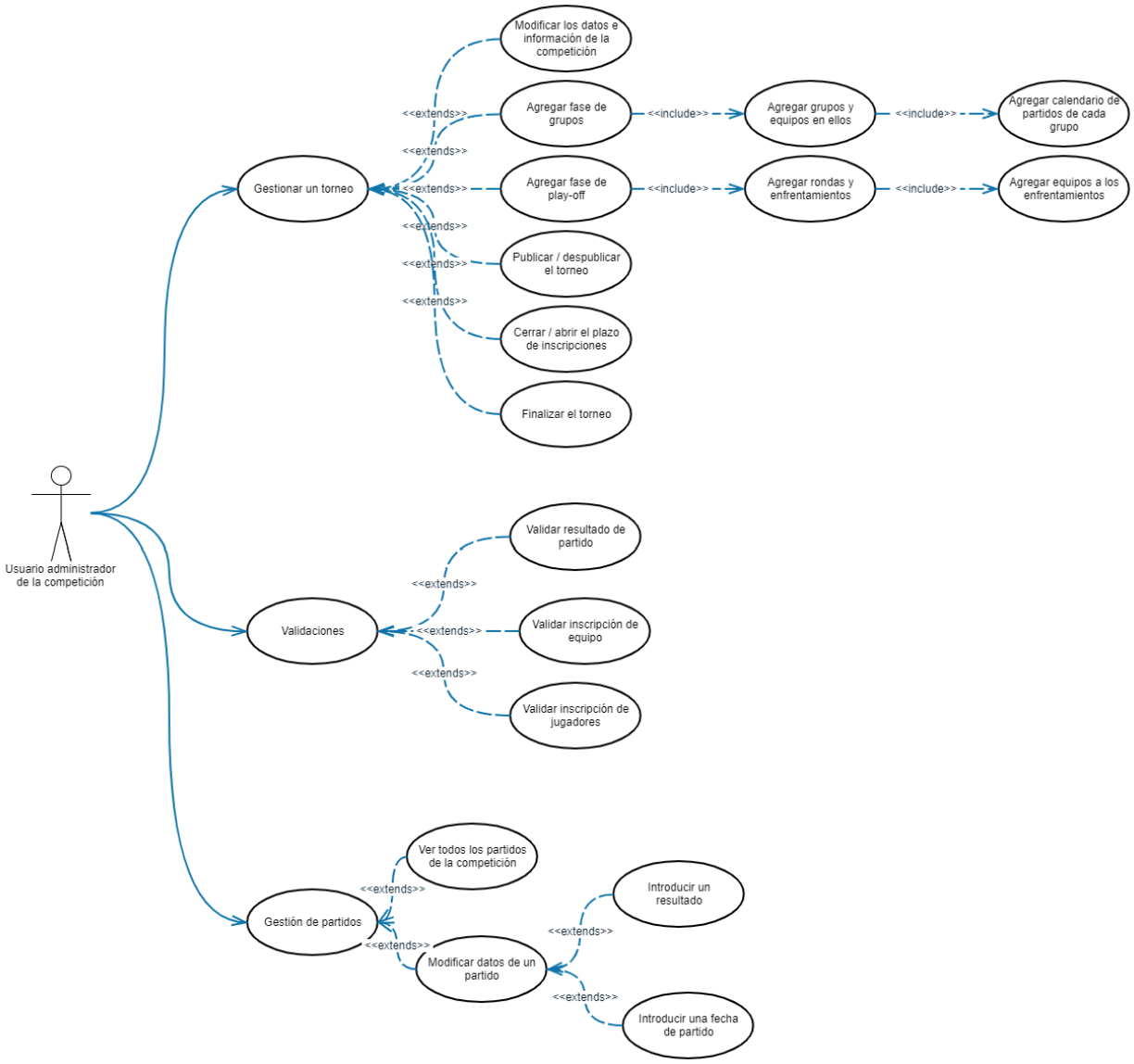
f.7 Caso de uso: usuario no registrado

2.7.2.2. Usuario administrador del sistema



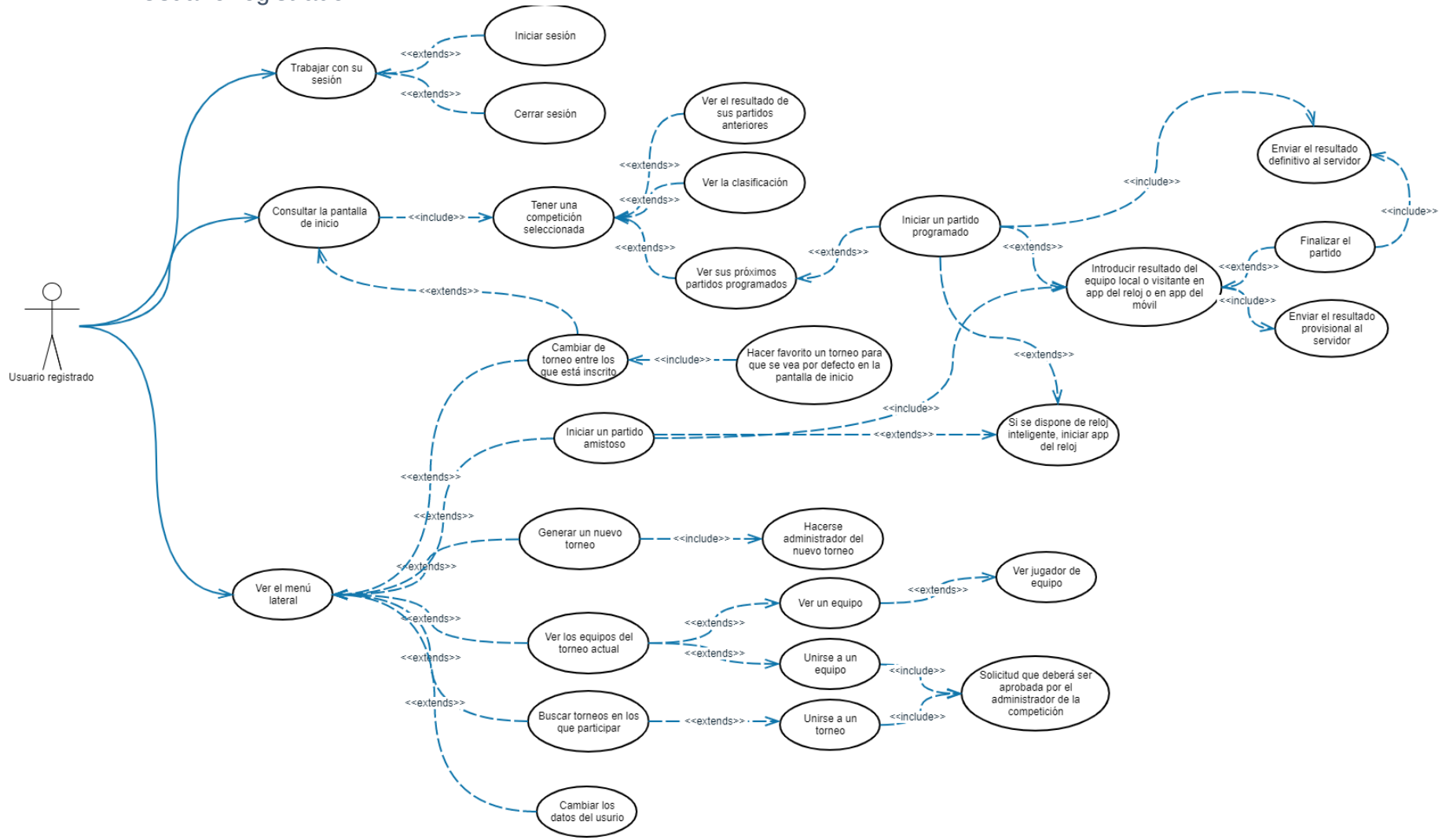
f.8 Caso de uso: usuario administrador del sistema

2.7.2.3. Usuario administrador de competición



f.8 Caso de uso: administrador de la competición

2.7.2.4. Usuario registrado



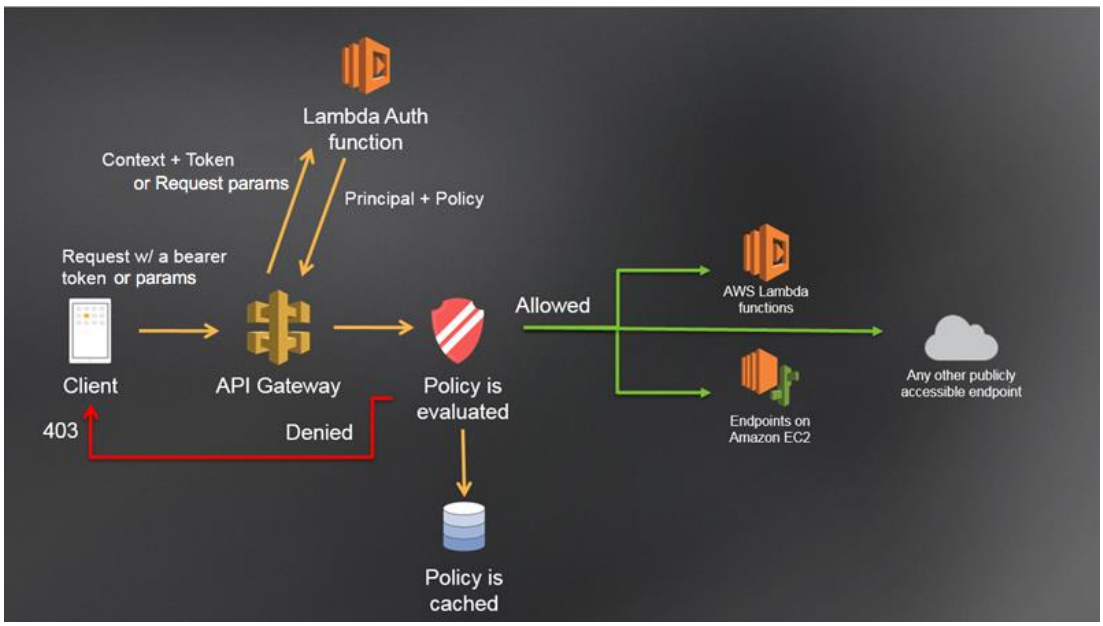
f.9: Caso de uso: Usuario registrado

2.8. Servidor de la aplicación

Actualmente disponemos de varios medios para poder trabajar con servidores de aplicaciones. Tecnológicamente podemos hablar de aplicaciones cliente-servidor monolíticas, que son aquellas en la que todo el backend se engloba detrás de una gran aplicación desplegada en un servidor tipo Tomcat con un Apache escuchando las peticiones. Desde hace unos años se empezó a ver la necesidad de fraccionar esas grandes aplicaciones para que fuesen más mantenibles y empezaron a aparecer los "microservicios", que no dejan de ser otra más más que pequeñas aplicaciones corriendo sobre servidores, dedicadas específicamente a una tarea. Dentro de esta tendencia a encapsular y minimizar las funcionalidades de una aplicación, nacieron los "Docker", que son contenedores que simulan un microsistema operativo que permiten contener un microservicio, y ellos mismos levantase, y propagarse para dar servicio según la demanda de peticiones. Hoy en día se está implantando una nueva tendencia llamada "serverless" que consiste en olvidarse por completo de los servidores de aplicaciones y sus contenedores y delegar toda la infraestructura en terceros. Podemos encontrar Firebase de Google, Azure de Microsoft o AWS de Amazon como grandes enclaves tecnológicos que ofrecen serverless.

Así pues, debido que este proyecto toca tecnologías actuales, y nace con la necesidad de diferenciarse de un mercado copado, ofreciendo novedades tecnológicas, se ha decidido hacer una aplicación con servidores "serverless", y por su potencia en el mercado y por la abundante y muy buena documentación de que dispone, para este propósito se ha optado por elegir a Amazon y AWS.

El esquema que proporciona AWS para usar su servicio "serverless" es el siguiente: [5]



f.10 Diseño de una petición a AWS

Para realizar esta aplicación se ha seguido este documento de Amazon donde se explican todas las partes de un servicio "serverless"

https://aws.amazon.com/es/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/

Dentro de este entramado de productos podemos destacar

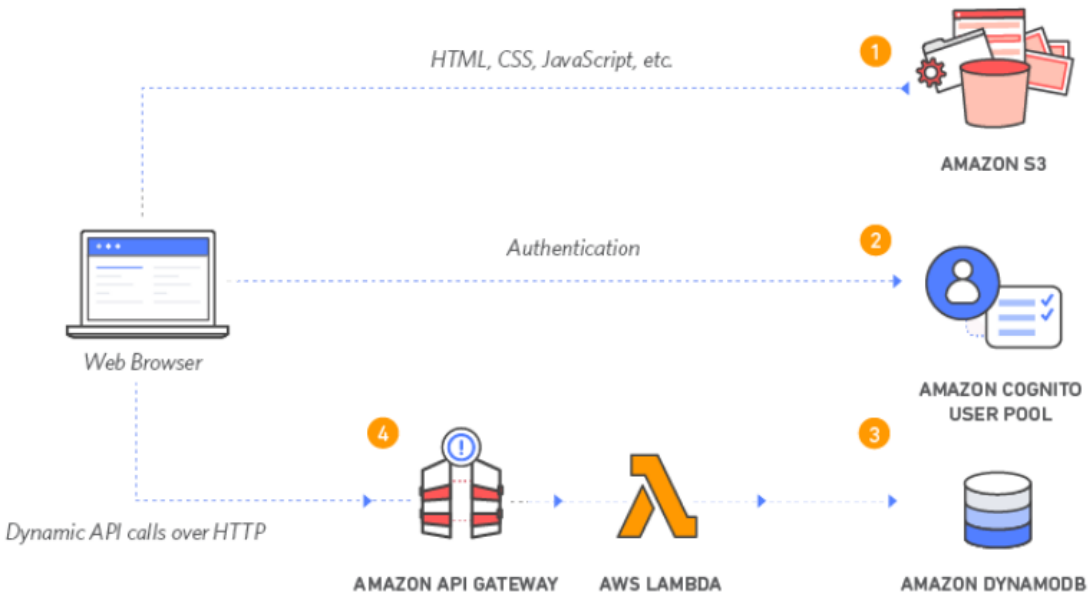
1. Amazon S3
   Servidores de Amazon dedicados al alojamiento tradicional de archivos, con servidores de aplicaciones y web corriendo en ellos, escalables y con un altísimo grado de disponibilidad
2. Amazon Cognito User Pool
   Servicio de Amazon destinado a la autenticación y autorización de usuarios.
3. Amazon DynamoDB

   Base de datos de tipo NoSql
4. Amazon API Gateway

   Interfaz de llamadas Rest para enrutarlas a las funcionalidades
5. AWB Lamdba

   Gestor de código “serverless” que ejecuta código no compilado

Estructura que puede quedar mejor reflejada en esta otra imagen:



f.11 Diseño del back-end según AWS

2.9. Objetivos de seguridad de la aplicación.

2.9.1. Teoría de seguridad

En una aplicación que utilice un api de consulta, basada en microservicios como herramienta de comunicación, es necesario tener en cuenta la seguridad. Para ello, se han estudiado las diferentes opciones que ofrece la tecnología actual para implementar la seguridad en las comunicaciones.

Las opciones que más fuerza tienen hoy en día son [3]:

- OAuth 2.0
- OpenID Connect
- JWT (Json Web Tokens)

Autenticación y autorización

Según “returngis.net” [3]: “La autenticación es el proceso de verificar una identidad, es decir confirmar que una persona es quien dice ser.”. “Por otro lado, la autorización es el proceso de verificar lo que un usuario puede hacer”.

La aplicación a desarrollar tendrá así pues estas dos fases de “autenticación” y “autorización”.

Dentro de las opciones que podemos ver, está OAuth, que se construyó específicamente para acceder a APIs a través de HTTP. OAuth es un framework para la autorización, que no la autenticación.

Por otra parte, OpenID Connect está pensado para la autenticación. [9] Y es un protocolo que implementa OAuth 2.0

Por último, encontramos JSON Web Token, el cual es un estándar, que pretende transmitir de manera segura información entre dos partes en formato JSON. Esta información puede ser verificada, ya que normalmente está digitalmente firmada.” [4]

La estructura de los tokens en este formato consiste en tres partes separadas por puntos, algo parecido a esto: xxxxxxx.yyyyy.zzzzz. Cada parte significa lo siguiente [3]:

- La cabecera, la cual normalmente tiene dos partes: el tipo de token y el algoritmo que se ha utilizado para ser firmado. Este puede ser HMAC SHA256 o RSA.
- El payload, esta es la parte que contiene los claims en Base64, es decir, los permisos de usuario.
- La firma, esta parte se utiliza para verificar que el mensaje no ha sido modificado durante el envío y, en el caso de los tokens que han sido firmados con una clave privada, además puede verificar que el emisor del token es quien dice ser.

2.9.2. Amazon Cognito

Una vez estudiada la tecnología, queda saber cómo se puede usar, y aquí es donde entra Amazon Cognito [10], un servicio de autenticación, administración y administración de usuarios, que implementa OpenID Connect, y usa JWT para verificar usuarios. Siendo, así pues, la mejor elección para realizar la aplicación.

2.10. Base de datos

A día de hoy nos encontramos con, principalmente, 2 tipos de formas de hacer una base de datos, a saber, las relacionales, y las NoSql (Not Only SQL)

En las primeras, la información se guarda en tablas, en filas con identificadores que permiten hacer relaciones entre tablas.

En las segundas, la información se guarda en documentos json, almacenados sin una estructura definida.

No se puede afirmar con rotundidad qué modelo es mejor, ya que para cada problema puede encajar mejor una u otra.

En una comparación simple, podríamos decir: [7] [8]

Una base de datos relacional es fácil de leer al tener la información organizada en tablas. Permite realizar consultas de unión entre varias tablas, y en general, almacenar una gran cantidad de información.

Una base de datos noSql permite organizar la información de forma más arbitraria, sin una estructura definida y cerrada. Es muy rápida para consultas de microservicios ya que devuelve objetos json ya formados sin tenerlos que parsear. Permite organizar información de cualquier tipo sin necesidad de generar tablas.

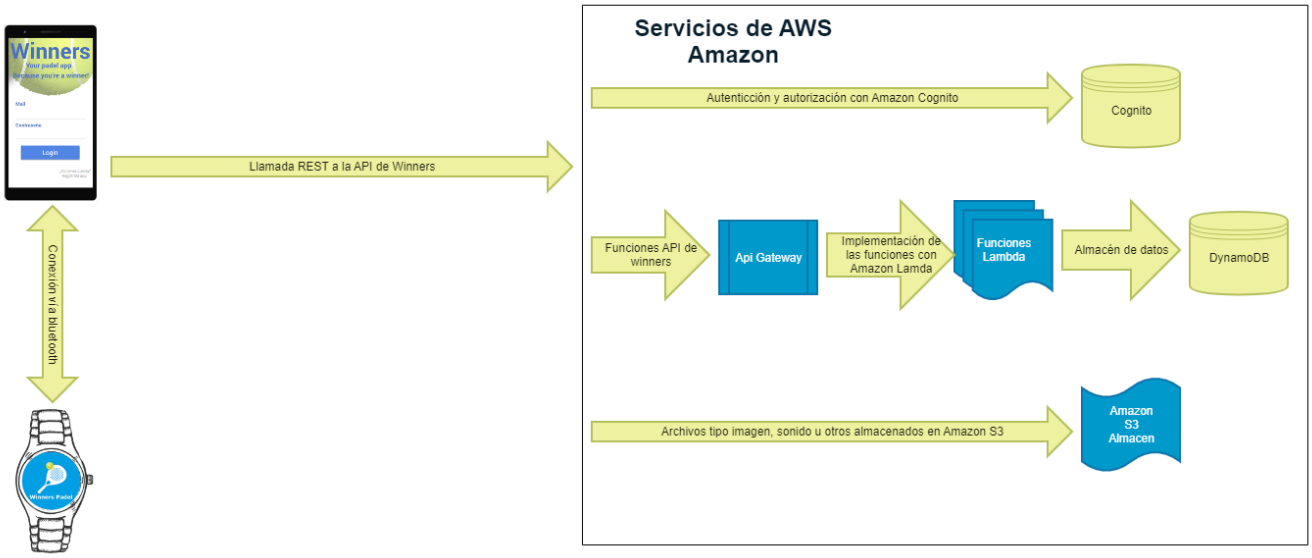
Es por ello, que para la aplicación he decidido almacenar la información en una base de datos noSql.

Los documentos json de la base de datos tendrán los atributos que se indican en el ANNEXO

2.11. Diagrama de implementación

Una vez diseñada la aplicación móvil y del reloj, y explicada la parte del servidor queda definir como se implementará la conexión entre todas las aplicaciones.

Para ello se ha definido el siguiente diseño de arquitectura física:

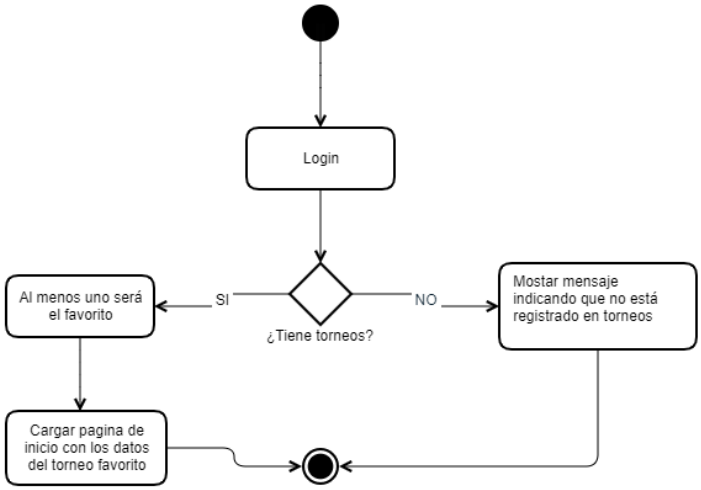


f.12 Diagrama de implementación

2.12. Diagramas de actividad

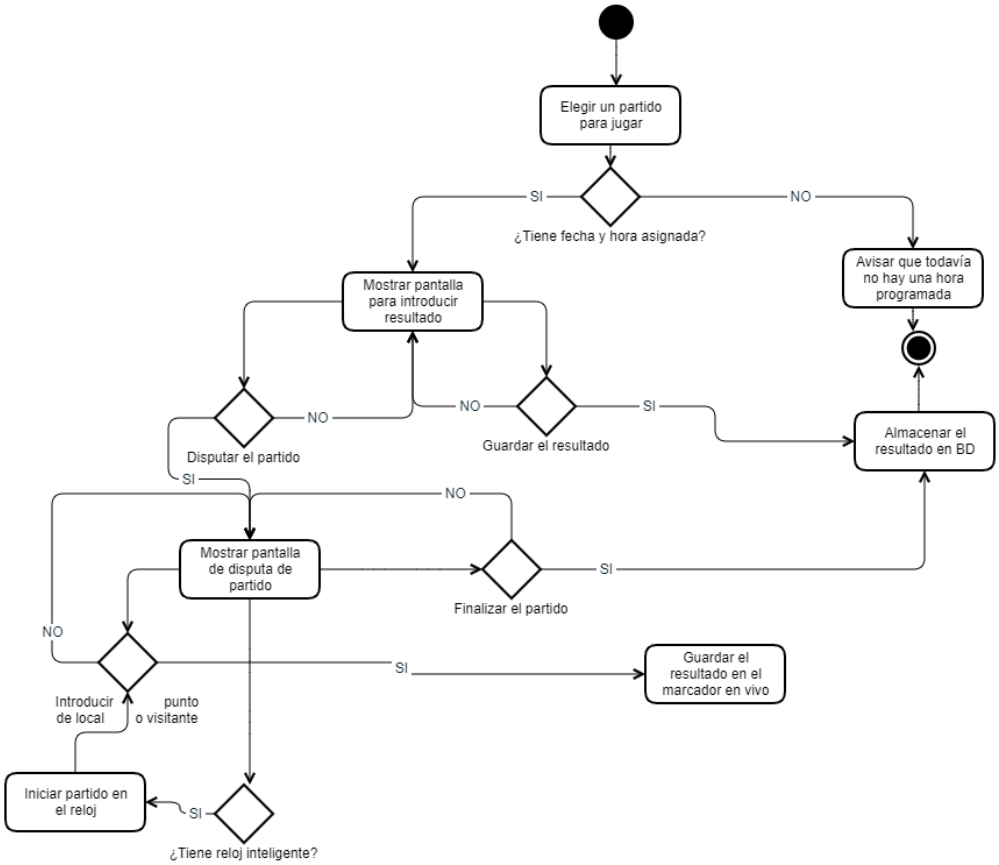
A continuación, pensando ya en el inicio del desarrollo, se formulan unos diagramas de actividad esenciales para entender partes básicas de la aplicación

2.12.1. Pantalla de inicio



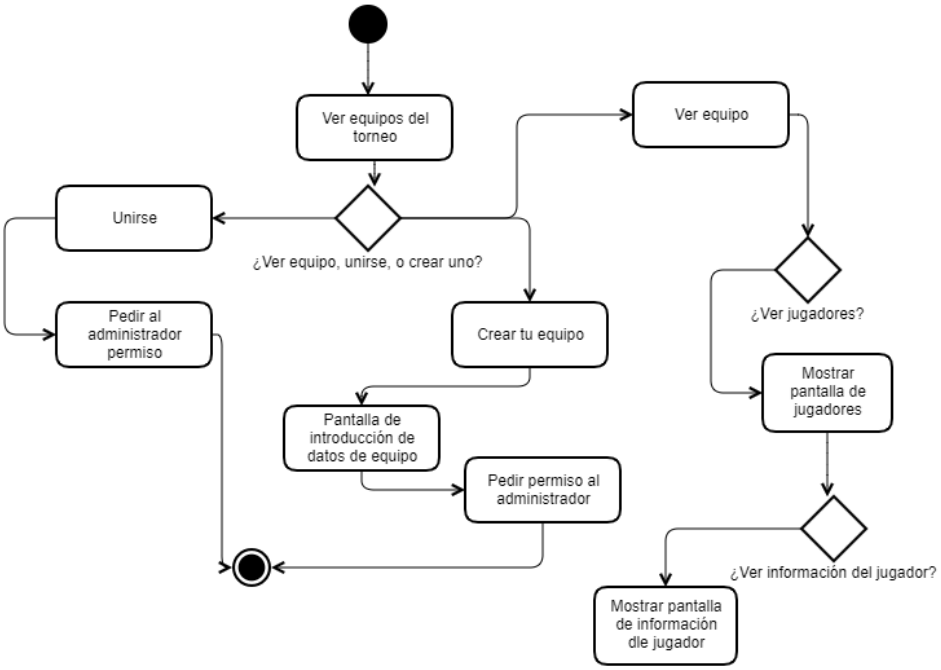
f.13 Diagrama de decisión: pantalla de inicio

2.12.2. Disputar un partido



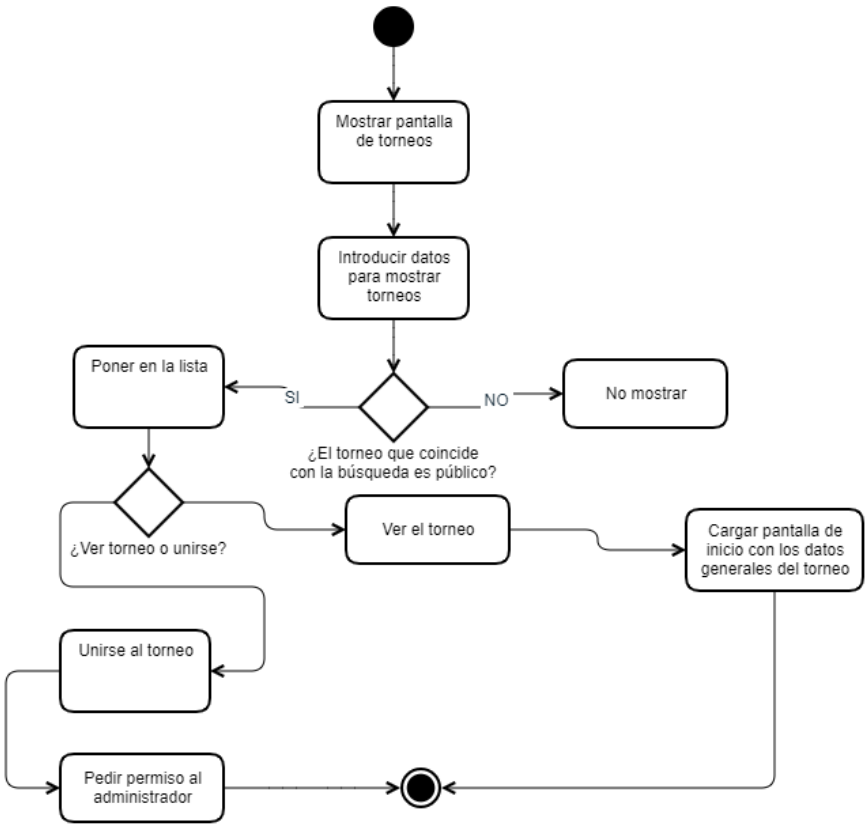
f.14 Diagrama de decisión: disputar un partido

2.12.3. Ver información de equipos e inscribirse en ellos



f.15 Diagrama de decisión: ver información de equipos e inscribirse en ellos

2.12.4. Ver torneos e inscribirse en ellos



f.16 Diagrama de decisión: ver torneos e inscribirse en ellos

3. Desarrollo

Una vez realizado todo el diseño de la aplicación se comienza el desarrollo de la programación.

Este desarrollo se realizará en varias fases. Se verá el backend y como se ha desarrollado todo con AWS.

Se entrará a valorar cómo se resuelto todas las consultas al backend y que problema nos hemos encontrado.

Veremos también como ha estado esto a la parte del frontend, y qué principios de arquitectura se han sabido además de todos los programas encontrados, lo que ha llevado a realizar cambios en el diseño y recortes en la planificación.

Por último, se terminará mostrando los diagramas más importantes de la aplicación

3.1. Backend: AWS

Para realizar el backend se ha decidido usar los servicios web que proporciona Amazon un mediante su tecnología AWS.

Gracias a los servicios Serverless, se puede desarrollar toda la parte del servidor sin tener que estar valorando sobre qué tecnologías trabajar, cómo ejecutarlas o cómo desplegar aplicaciones en ellas.

AWS ofrece una gran cantidad de servicios en la nube [11], de los cuales se van a usar para desarrollar la aplicación los siguientes:

- **S3**, para alojar contenido estático.
- **Route53**, como proxy redireccionador de llamadas.
- **CloudFront**, servicio usado para usar la certificación de Amazon y realizar así llamadas seguras
- **Certificate Manager**, para usar la entidad certificadora oficial de Amazon y generar un certificado del dominio.
- **Cognito**, para la identificación y logueo de usuarios, así como para cifrar las comunicaciones mediante JWT.
- **Api Gateway**, para crear el frontal con el api, donde ejecutar las llamadas desde la aplicación móvil.
- **Lambda Functions**, como servicio serverless en la nube que ejecutará todo el código desarrollado para recibir las llamadas y realizar consultas a la base de datos.
- **DynamoDB**, como base de datos elegida el formato NoSQL, ya que brinda una serie de características apreciadas para realizar aplicaciones móviles como son la velocidad y la facilidad en la consulta de datos.
- No se debe olvidar tampoco el servidor de domino DNS, Nominalia, al cual se le compró el dominio de la web.
- Así como Google Analytics que servirá para poder dimensionar correctamente la acogida de la aplicación.

Todos estos servicios de AWS se encuentran distribuidos a nivel mundial en diferentes ciudades de diferentes países.

El data center elegido para alojar toda la infraestructura ha sido el de Fráncfort.

Dividiremos en 2 grupos los servicios. Los de “Sistemas” y los de “DevOps”

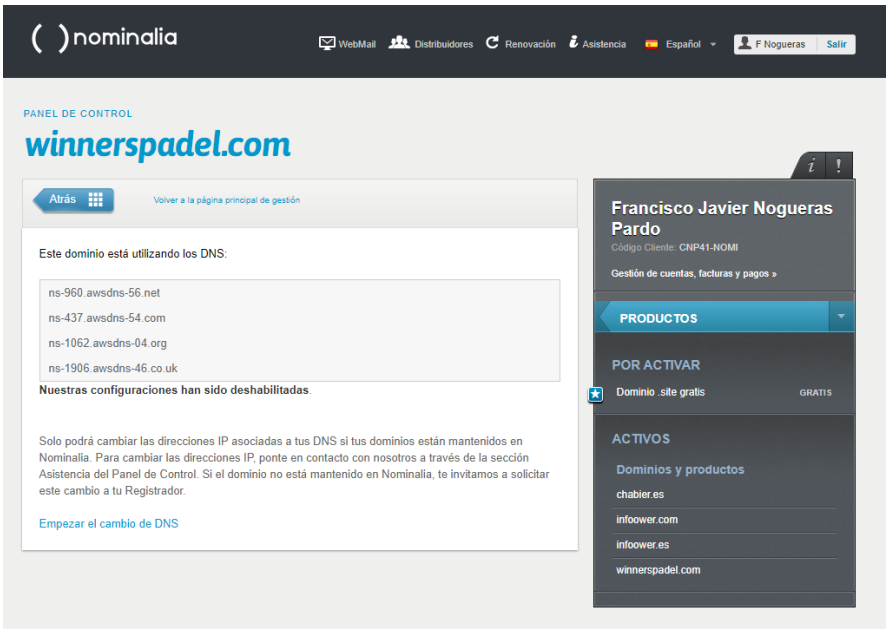
3.1.1. Sistemas

Se incluirán como servicios de Sistemas, todos aquellos que no dependan directamente de la programación de la aplicación, siendo el soporte fundamental en el que todo el backend se basará.

3.1.1.1. Nominalia

El primer paso fue conseguir un nombre de dominio para poder realizar las llamadas sin tener que depender de rutas que con el tiempo podrían cambiar, o quizá ser bloqueadas por firewalls.

Es por ello que se adquirió “https://winnerspadel.com/”

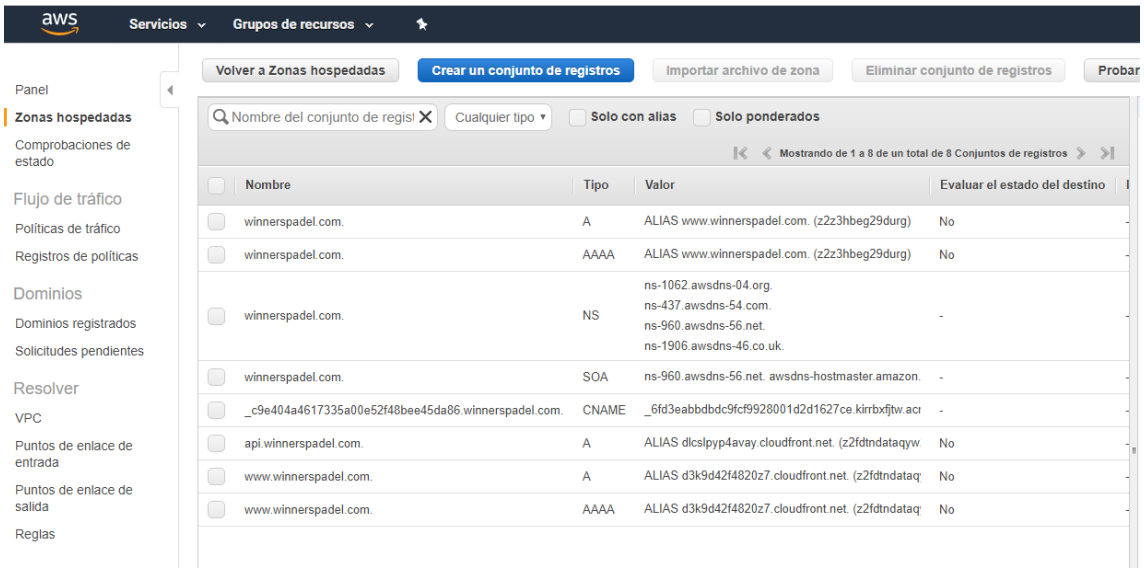


f.17 Página de Nominalia

Donde se configuraron los DNS ofrecidos por Amazon en la herramienta Route53

3.1.1.2. Route 53

Servicio que tras la configuración y redirección de DNS nos conectará nuestro dominio de Nominalia, con todos los servicios de Amazon.

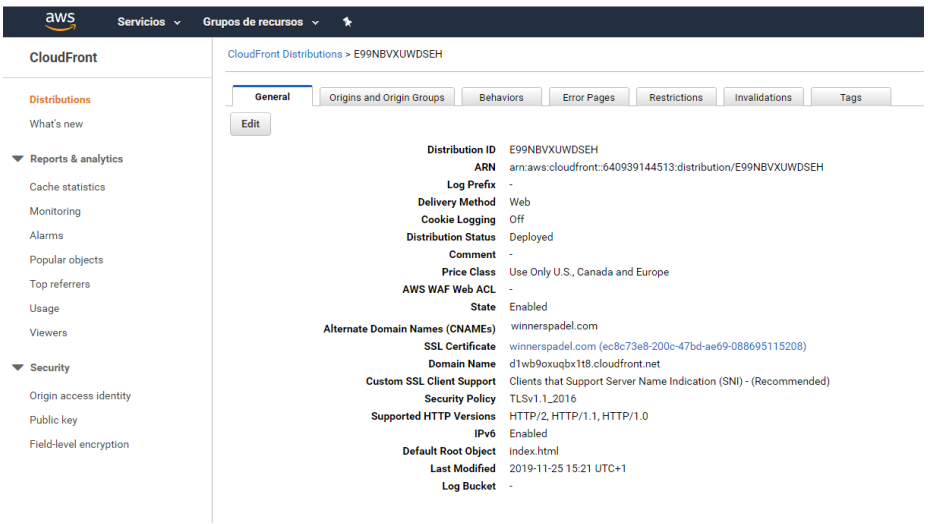


f.18 Página de AWS, Route 53

Como se puede ver en la imagen, se tuvo que realizar una configuración bastante detallada de muchas redirecciones de tipo A,AAAA,NS,SOA y CNAME

3.1.1.3. CloudFront

Con este servicio, conectado con Route53, conseguimos que se valide una conexión segura por HTTPS



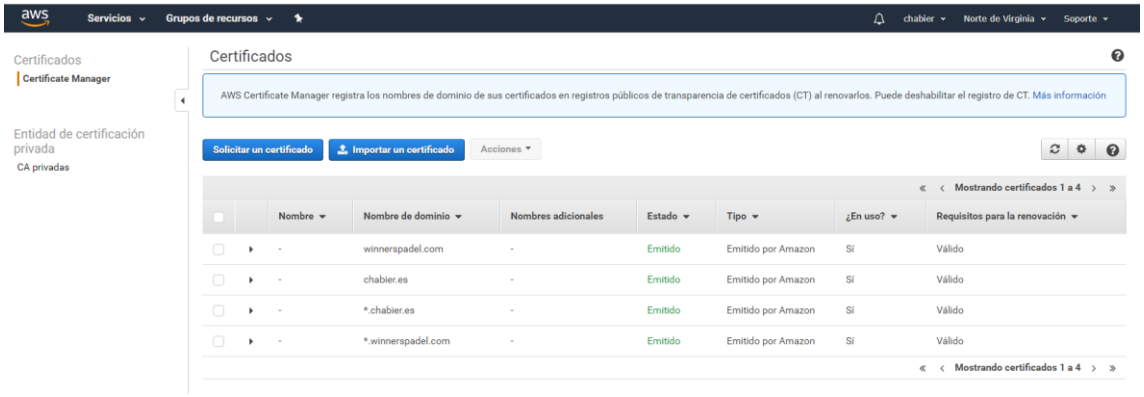
f.19 Página de AWS, CloutFront

La configuración una vez más, detallada y minuciosa con muchos apartados llevó bastante tiempo.

3.1.1.4. Certificate Manager

Amazon es una entidad certificadora oficial, por lo que se pueden pedir certificados gratuitos para sus servicios.

Certificados que se enlazarán con CloudFront y Route53 para poder validar las conexiones https a cualquier punto de la aplicación.

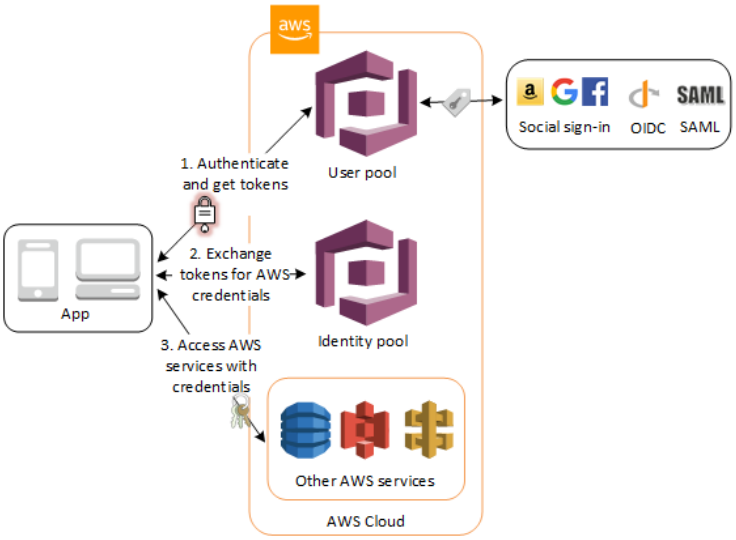


f.20 Página de AWS, Certificate Manager

3.1.1.5. Cognito

Cognito empieza a salir del ámbito de “sistemas” pero debido a que una vez configurado, ya puede ofrecer un servicio ajeno a la programación, se engloba también este ámbito.

Para configurar Cognito Amazon nos guía, gracias a su fabulosa documentación. [25]



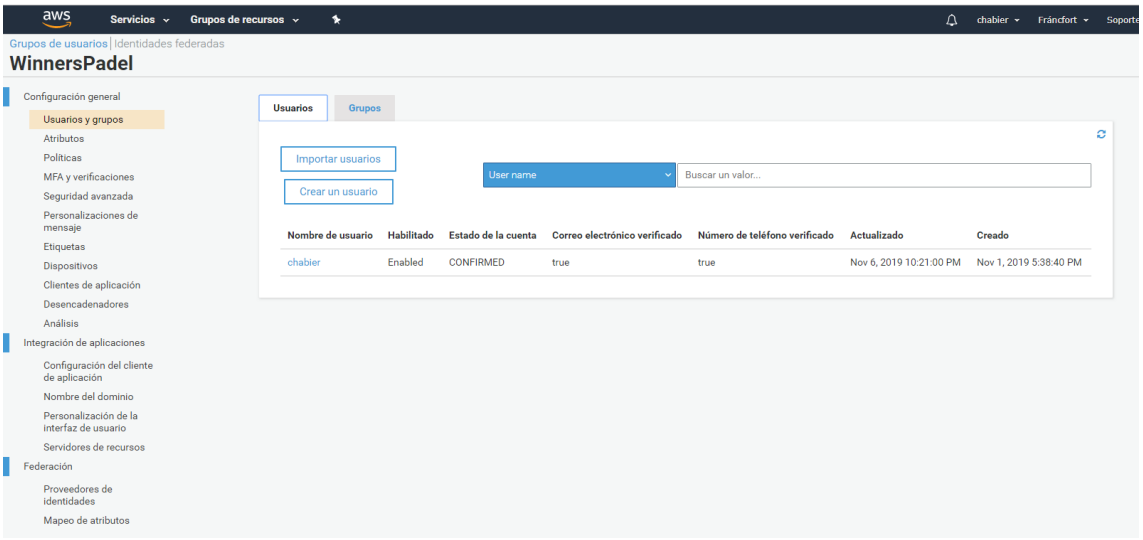
f.21 Diseño de la infraestructura de cognito

Como podemos ver en la gráfica superior, gracias a Amazon Cognito, podemos desarrollar un sistema para identificar los usuarios y asociar estos a una aplicación.

Aunque inicialmente no parece una lógica complicada y Amazon dispone de muy buenos documentos para poder realizar la implementación, esta no es tan asequible como inicialmente parece cuando se intenta desarrollar algo que se escapa de un ejemplo más sencillo.

El siguiente vídeo enlazado, o grupo de videos si se navega entre ellos, se muestra un tutorial que ayuda de forma bastante satisfactoria a poder realizar la conexión con Amazon Cognito. Quiero destacarlo porque tras muchas lecturas, y muchos tutoriales, este video [26] fue el que terminó de ayudarme en mi caso específico.

Cognito ofrece una base de datos donde almacenar los usuarios pudiendo darles permisos a estos para acceder a la aplicación

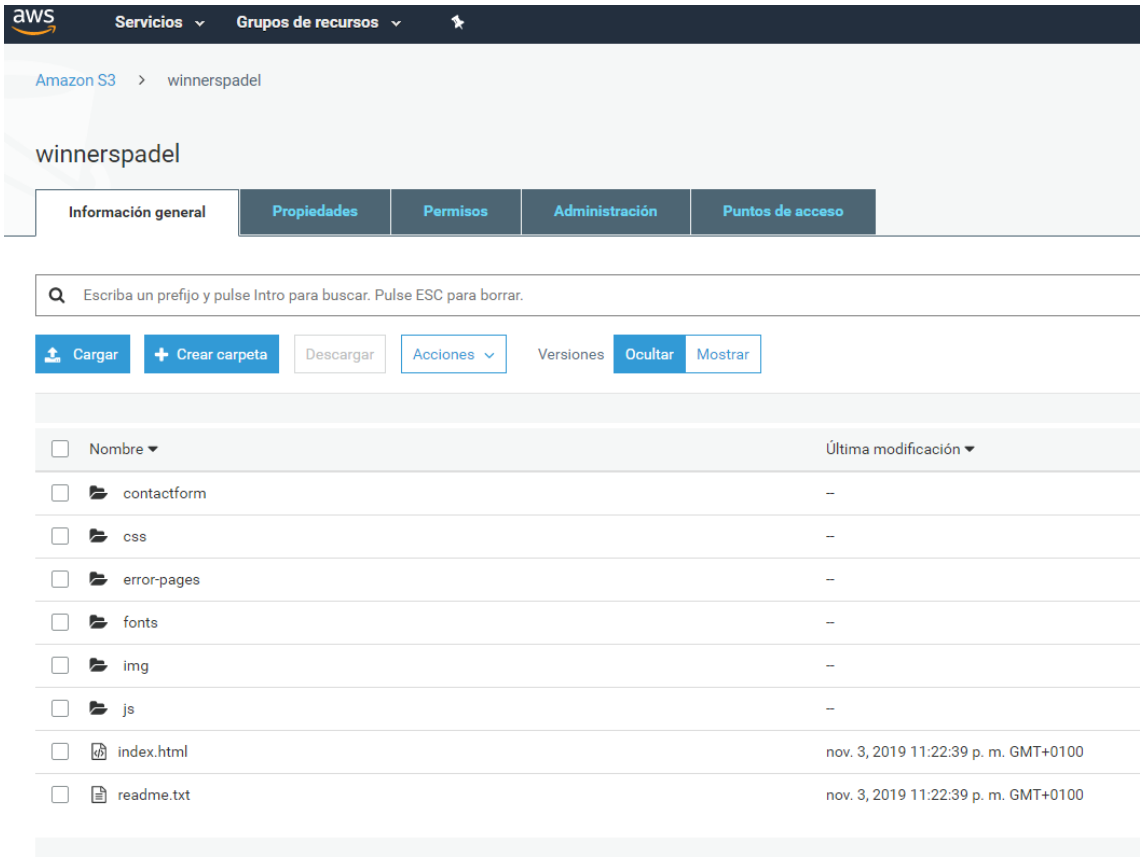


f.22 Página de AWS, Cogito

El usuario, una vez identificado en Cognito recibirá un token JWT para poder seguir usando el resto de servicios de la API de forma segura.

3.1.1.6. S3

Los servidores de S3 alojan contenido estático como puede ser una web, fotos, o recursos que se quieran usar desde la aplicación.

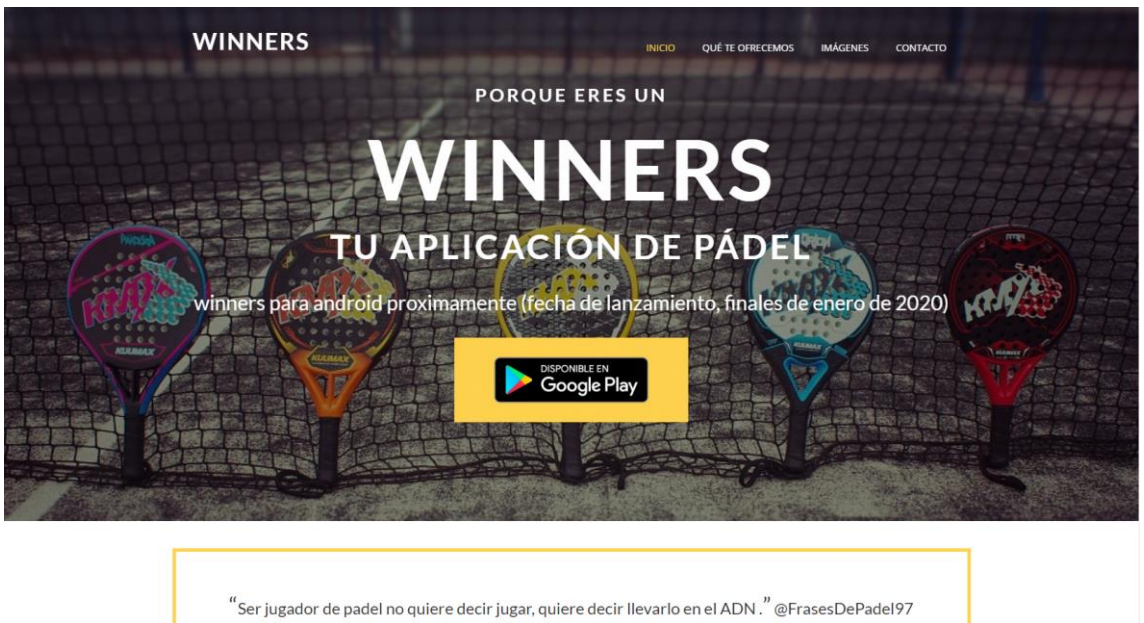


f.23 Página de AWS, S3

Una vez comprado el dominio y realizadas todas las redirecciones se aprovechó para realizar una web identificativa de la aplicación, que aunque no entre en los planes de este proyecto, se valoró como positiva.

Para optimizar lo más posible el tiempo disponible se decidió usar una plantilla sacada de BootstrapMade.

La web desarrollada es la siguiente:

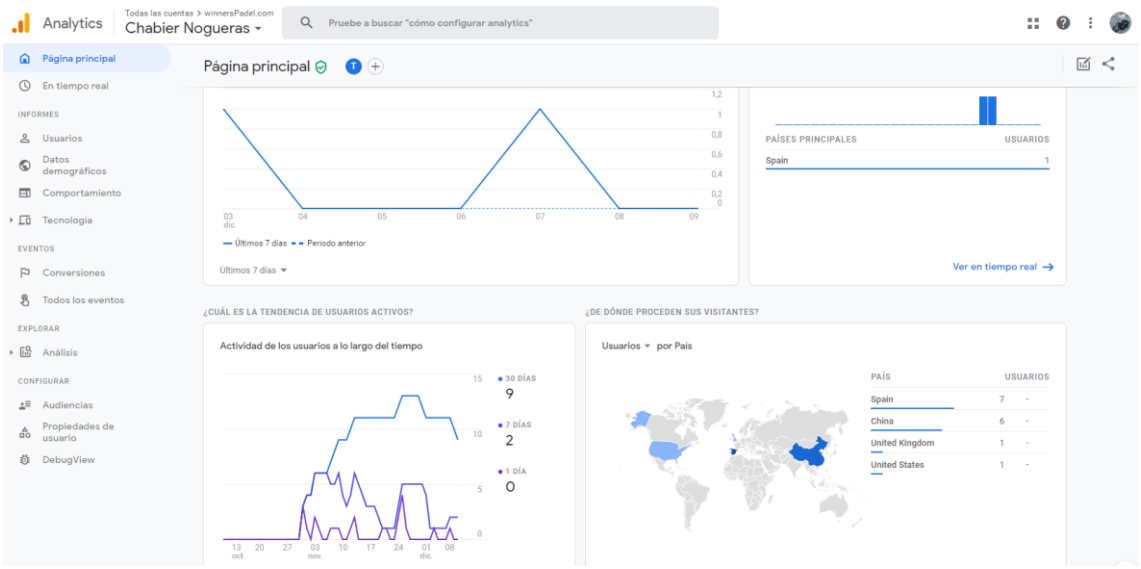


f.24 Página de WinnersPadel.com

3.1.1.7. Google Analytics

Por último, pero también importante, es poder conocer qué volumen de tráfico y de usuarios podemos esperar en la aplicación.

Para ello, ya que Amazon no ofrece ninguna herramienta que pueda gestionar esta información, al menos por ahora, se decidió por usar la herramienta de Google “Analytics”



f.25 Página Google Analytics

3.1.1.8. Problemas encontrados en la parte de Sistemas.

Todos los problemas que aquí se han ido resolviendo, son básicamente la falta de formación y conocimiento en esta herramienta que es bastante novedosa en el mundo informático español.

Como esta parte de sistemas no entra dentro de la misión de este proyecto, no nos extenderemos en enumerar todos los problemas que hubo que afrontar, aunque estos se podrían resumir en la conexión de todos los servicios entre ellos para poder disponer de un backend robusto y seguro.

3.1.2. DevOps

En esta parte se englobarán todos los servicios que de forma más directa trabajan con el código de la aplicación y dan servicio a nivel de programación a toda la app móvil.

3.1.2.1. Apigateway

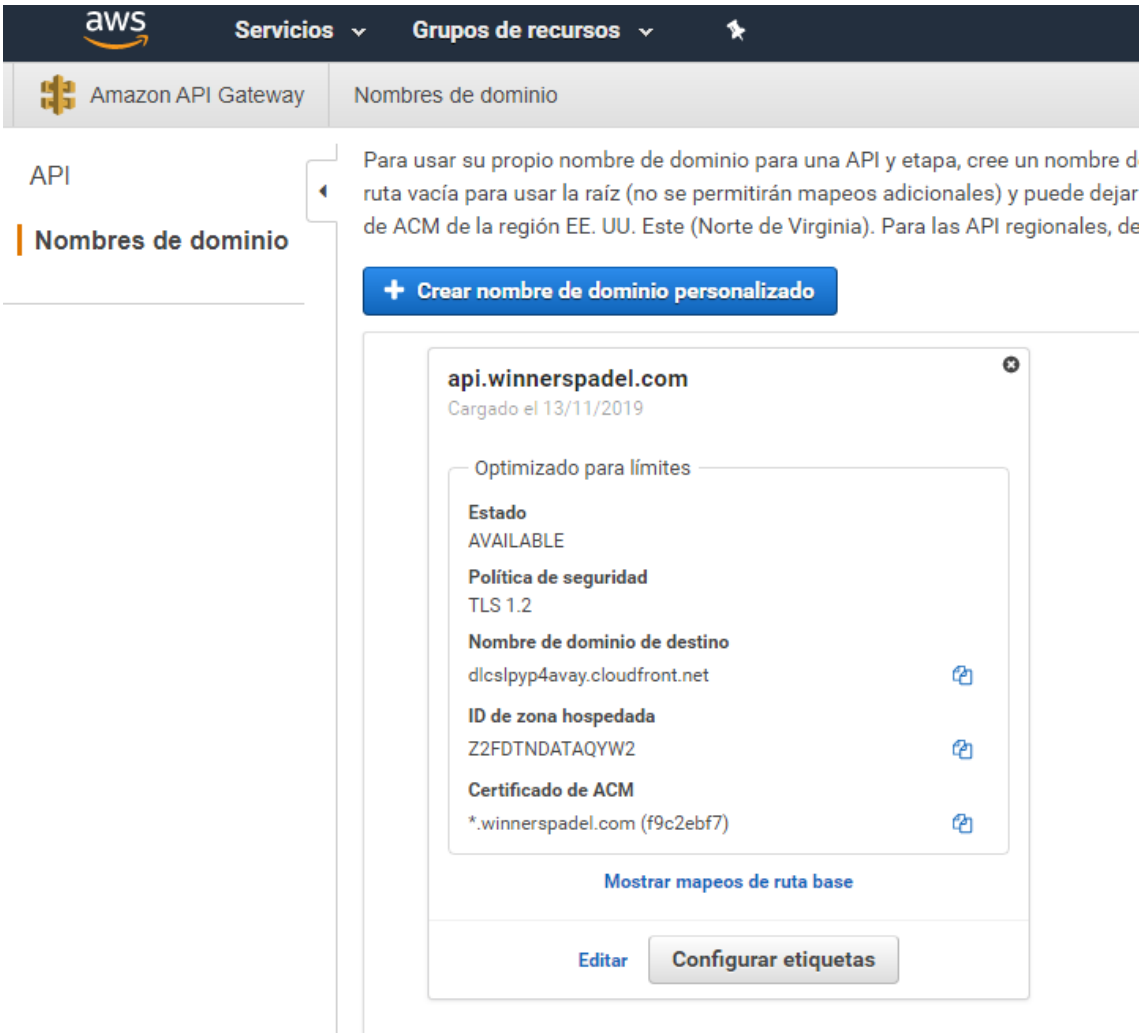
Punto de entrada en la aplicación. Podemos consultar en Amazon en qué consiste

f.26 Diseño de la infraestructura según API Gateway

Desarrollar la parte del API Gateway no ha resultado sencillo.

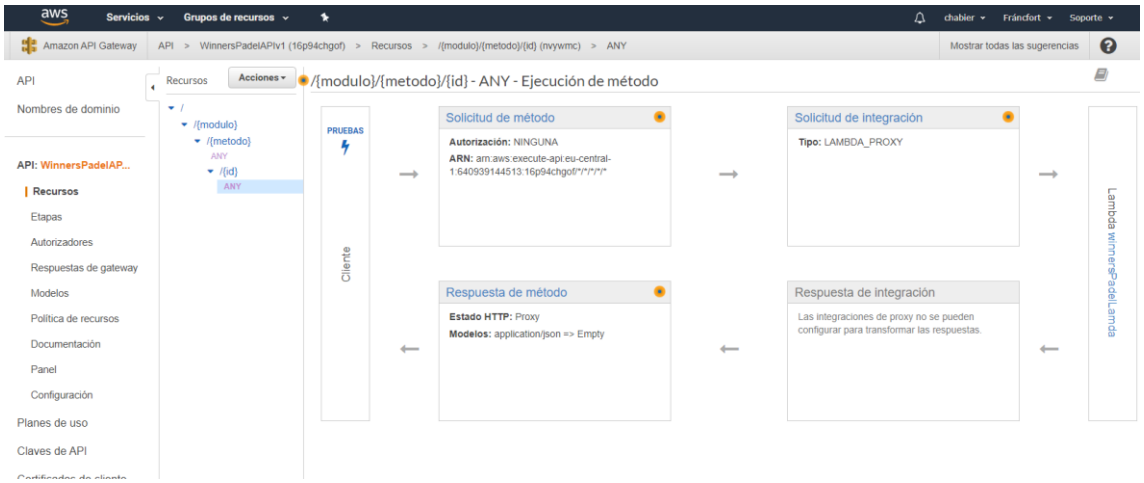
Hay que configurar bien 2 apartados importantes.

El dominio, con toda la configuración anterior



f.27 Página de AWS, API Gateway

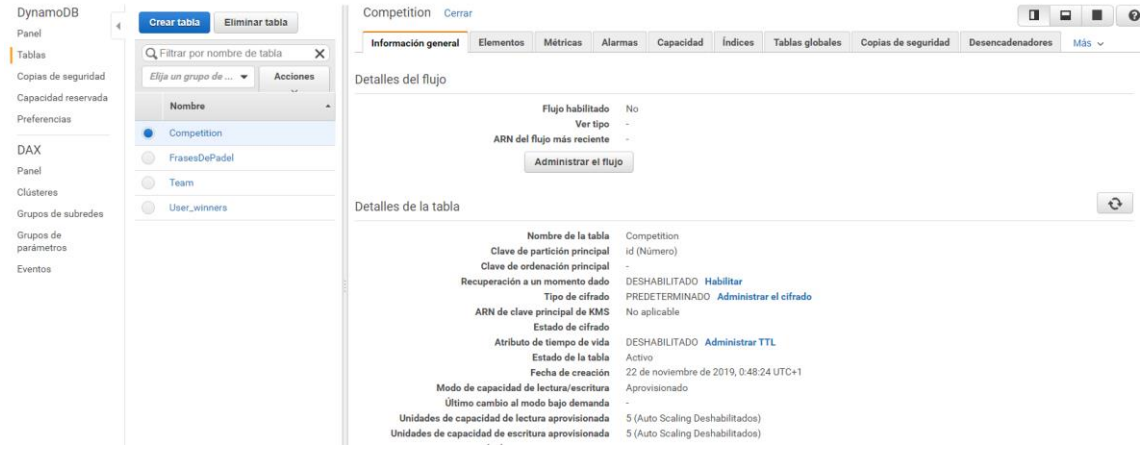
Y la API Rest que dará servicio a la aplicación.



f.28 Página de AWS, Métodos y recursos en API Gateway

3.1.2.2. Dynamodb

DynamoDB es la solución que ofrece Amazon a las bases de datos no relacionales de tipo NoSQL en competencia directa con MongoDB



f.29 Página de AWS, Dynamo DB

En ella se aloja toda la información de la aplicación, la cual se tuvo que dividir en 4 elementos principales.

Esta división se ha pensado basándose en “objetos primarios” de la aplicación.

Una tabla para documentos de tipo “Competición” donde cada documento guardará una competición entera, con todas sus fases, equipos y usuarios.

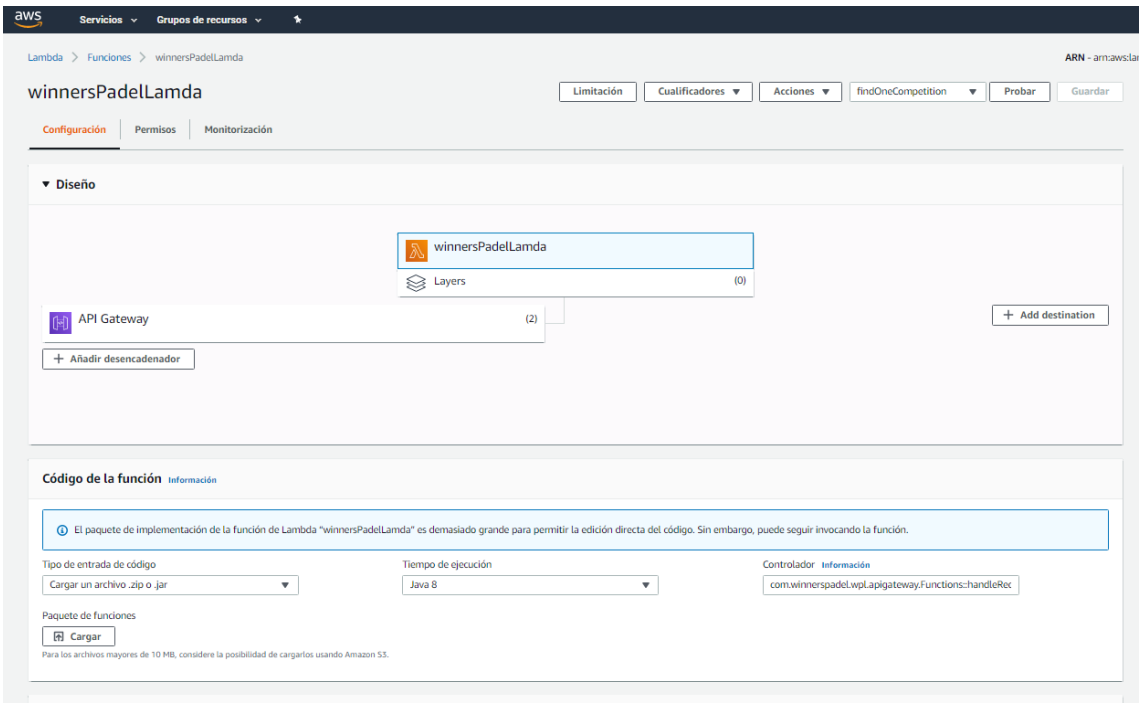
Una tabla con documentos de tipo “Team” donde se guardarán todos los equipos generados en todas las competiciones.

Una tabla para almacenar los “User_Winners” que darán forma a cada uno de los usuarios de la aplicación.

Y, por último, agregada a última hora, una tabla con “frasesDePadel” que almacena documentos con frases para mostrar en la pantalla de carga de la app.

3.1.2.3. Lambda

Esta es la parte principal de todo el desarrollo del backend.



f.30 Página de AWS Servicios Lambda

Este contenedor es lo que se denomina “función lambda”, y en él solo se sube un jar con el código, en caso de aplicaciones java como la desarrollada.

Nos olvidamos por completo de mantener servicios, mantener hardware, mantener aplicaciones, mantener instancias, etc.

AWS “despliega” la aplicación con cada llamada que se realiza a él.

Para ello, se tiene 2 tipos de arranque

- Cold Start, donde AWS tiene que levantar desde 0 toda la aplicación, y lo cual tiene un tiempo de espera que, dependiendo de la aplicación, puede ser más o menos largo.
- Warm Start, una vez que la aplicación está levantada, AWS la mantiene viva durante unos pocos minutos más para poder dar servicio al resto de llamadas de una forma más rápida

Debido al “Cold Start” se tuvo que hacer en la parte móvil una pantalla de carga de la aplicación donde se pedían asíncronamente todos los datos mientras AWS iniciaba la función Lambda.

La aplicación que se entrega a Lambda para que la gestione se llama

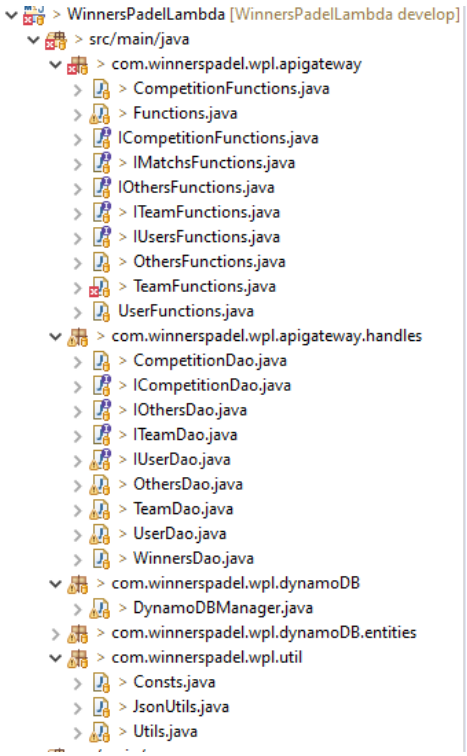
“WinnersPadelLamba”

La estructura que se ha diseñado en la aplicación ha estado muy ligada al desarrollo de API Gateway.

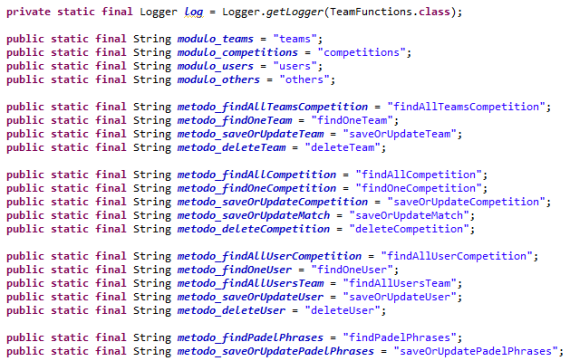
Como se explica en el apartado 1.1.2.4 de “problemas”, el desarrollo fue evolucionando viendo las necesidades que se planteaban.

Así pues, se terminó diseñando una aplicación con una clase de entrada principal a la función lambda:

```
public class Functions implements RequestStreamHandler
```



A partir de ella se hizo una suerte de proxy, que recogía los valores de APIGateway para enrutar la petición a la función lambda diseñada.



```
private static final Logger log = Logger.getLogger(TeamFunctions.class);

public static final String modulo_teams = "teams";
public static final String modulo_competitions = "competitions";
public static final String modulo_users = "users";
public static final String modulo_others = "others";

public static final String metodo_findAllTeamsCompetition = "findAllTeamsCompetition";
public static final String metodo_findOneTeam = "findOneTeam";
public static final String metodo_saveOrUpdateTeam = "saveOrUpdateTeam";
public static final String metodo_deleteTeam = "deleteTeam";

public static final String metodo_findAllCompetition = "findAllCompetition";
public static final String metodo_findOneCompetition = "findOneCompetition";
public static final String metodo_saveOrUpdateCompetition = "saveOrUpdateCompetition";
public static final String metodo_saveOrUpdateMatch = "saveOrUpdateMatch";
public static final String metodo_deleteCompetition = "deleteCompetition";

public static final String metodo_findAllUserCompetition = "findAllUserCompetition";
public static final String metodo_findOneUser = "findOneUser";
public static final String metodo_findAllUsersTeam = "findAllUsersTeam";
public static final String metodo_saveOrUpdateUser = "saveOrUpdateUser";
public static final String metodo_deleteUser = "deleteUser";

public static final String metodo_findPadelPhrases = "findPadelPhrases";
public static final String metodo_saveOrUpdatePadelPhrases = "saveOrUpdatePadelPhrases";
```

El resto de estructura del proyecto se diseñó siguiendo los patrones de programación MVC

f.31 Imágenes del código del servicio Lambda

Así pues, encontraremos clases de “funciones” que realizarán la interacción con el exterior recogiendo datos, clases de tipo “DAO” que serán las que realizarán la consultada de datos pedidas desde las funciones, y clases de tipo “entity” que serán las que representarán el modelo de datos.

3.1.2.4. Problemas encontrados en la parte de DevOps

ApiGateway

El primer problema que afronté fue como diseñar la API Gateway. Ya que, debido al tipo de funcionalidad de Lambda, cada uno de los métodos tiene que apuntar solamente a una función.

Inicialmente se diseñó de forma que se estructuraba todos y cada uno de los métodos de la aplicación en rutas del tipo /teams/findOneTeam (GET), /teams/saveTeam (POST), /competition/findOneCompetition (GET) etc.

Este diseño, aunque muy claro, y quizá más correcto con pequeñas funciones lambda apuntando a cada uno de los recursos, se tuvo que descartar, ya que para la programación del backend se usó una sola aplicación Java de casi 10MB de peso y no se podía alojar esta de forma repetida en multitud de lambdas para dar servicio a cada uno de los métodos. Además, esto habría hecho inmanejable el desarrollo y ampliación de la aplicación.

Es por ello que se terminó diseñando un servicio basado en variables, como se puede ver en la imagen, que de forma genérica recogiesen el módulo y el método que se querían usar, y así en una única función Lambda, se usasen estas variables para ir a un método u otro.

DynamoDB

Aunque aquí los problemas han sido menos, si que han existido, sobre todo por desconocimiento de la herramienta.

Hubo que tipar con anotaciones todos los elementos de las “entity”.

```java
import java.io.Serializable;


@Data
@JsonInclude(JsonInclude.Include.NON_NULL)
@JsonIgnoreProperties(ignoreUnknown = true)
@AllArgsConstructor
@NoArgsConstructor
@DynamoDBTable(tableName = "Competition")
public class Competition implements Serializable {


  private static final long serialVersionUID = -5781245487418016232L;

  @DynamoDBHashKey(attributeName = "id")
  private Long id;
  private Long idUserCreated;
```

f.32 Entity en java, representando su relación con DynamoDB

Sin embargo, algunas de esas anotaciones tienen un tratamiento especial, como es el caso de los “ID”.

Dynamo obliga a marcar como @DynamoCBHashKey las claves primarias de los documentos principales.

Existe otra anotación llamada @DynamoDBAutoGeneratedKey Sin embargo ambas anotaciones son incompatibles si la clave primaria es un número, ya que la clave autogenerada es un hash.

Otra de las anotaciones que causó problemas son la de los objetos anidados @DynamoDBDocument ya que Dynamo no sabe reconocer un objeto dentro de otro, y hubo que anotar todos los objetos.

Como ya se ha indicado, el desconocimiento de la herramienta fue el principal problema.

LambaFunctions

Así los problemas se replicaron de los anteriores puntos. Cómo realizar consultas a Dynamo, y como recuperar y tratar los datos recogidos de la API.

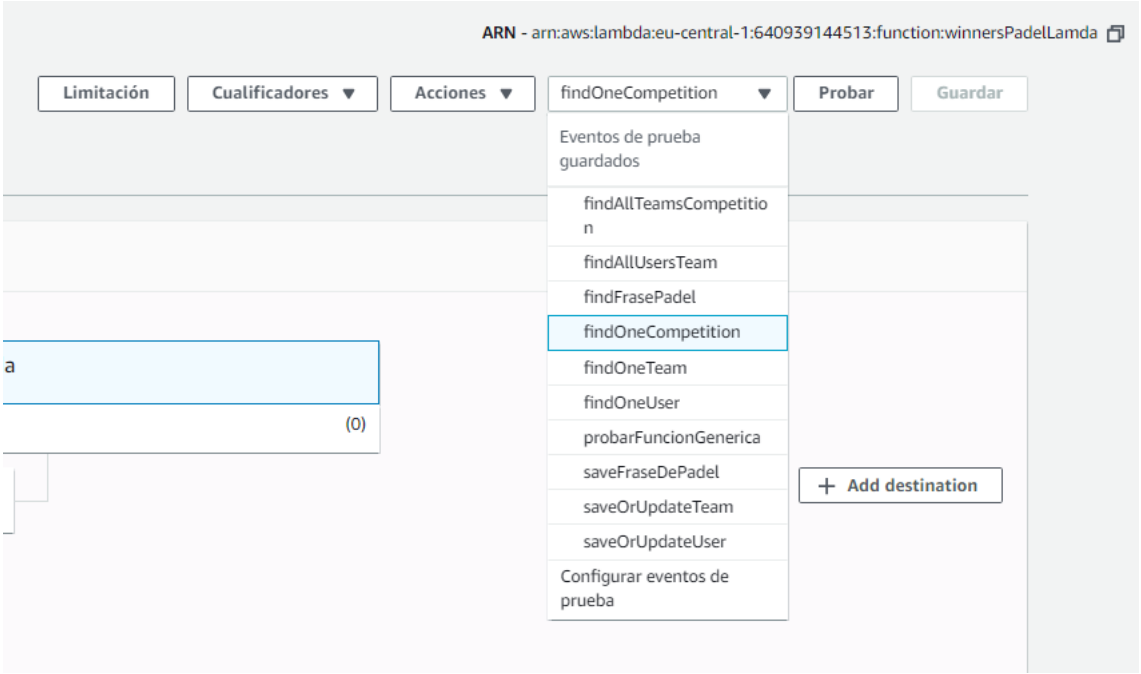
El principal problema que existe es que Amazon tiene hasta 3 formas diferentes de consultar datos a Dynamo. Con el objeto “DynamoDB” que trabaja con Items, con el objeto “AmazonDynamoDB” que ya solo con el nombre parecido confunde, y el objeto “DynamoDBMapper” que permite guardar objetos que el internamente mapea en Items.

Conocer, y saber utilizar estas 3 herramientas, que son una evolución unas de otras, ha sido el principal problema que he afrontado en esta parte.

3.1.2.5. Test de funcionamiento

Pruebas Lambda

Lambda permite realizar hasta 10 test de funcionamiento de la función para validar su desempeño.



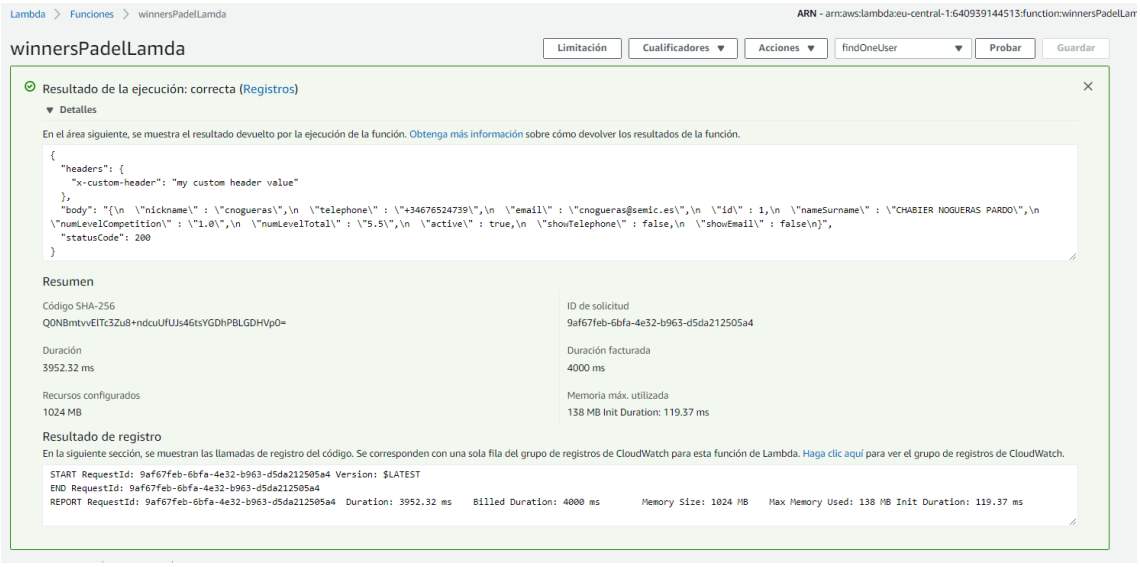
f.33 Pruebas en el servicio Lambda

Estas pruebas simulan una llamada real al servicio Lambda para verificar su funcionamiento

f.34 Json de una prueba para el servicio Lambda

En estas pruebas tenemos que indicar, entre todas las variables que se pueden poner, principalmente los “pathParameters” para comprobar la llamada al módulo/método correcto y, si así la función lo requiere, el “body” con el objeto que se quiere pasar a la función

Una vez invocada tenemos este resultado

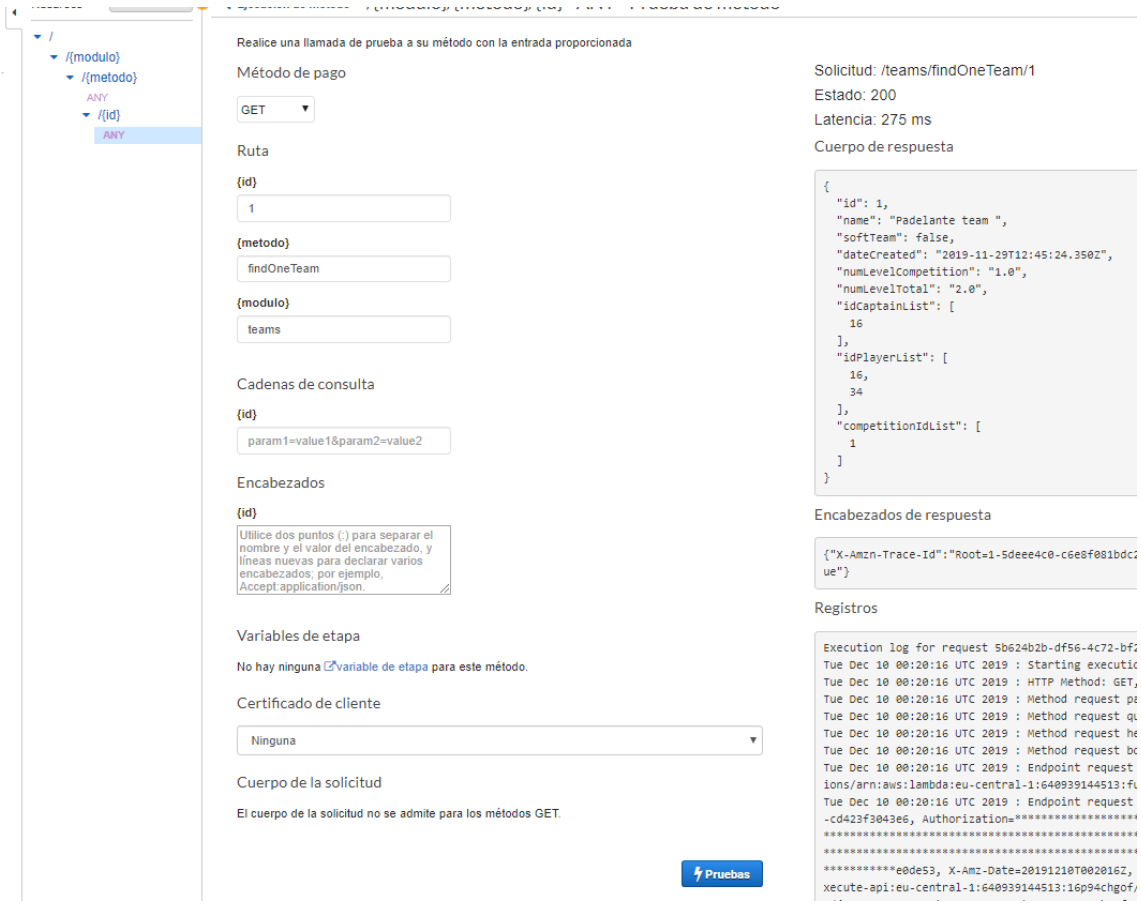


f.35 Resultado de una prueba del servicio Lambda

Pruebas “ApiGateway”

Por otra parte, nos encontramos con las pruebas desde el siguiente nivel que es “ApiGateway”

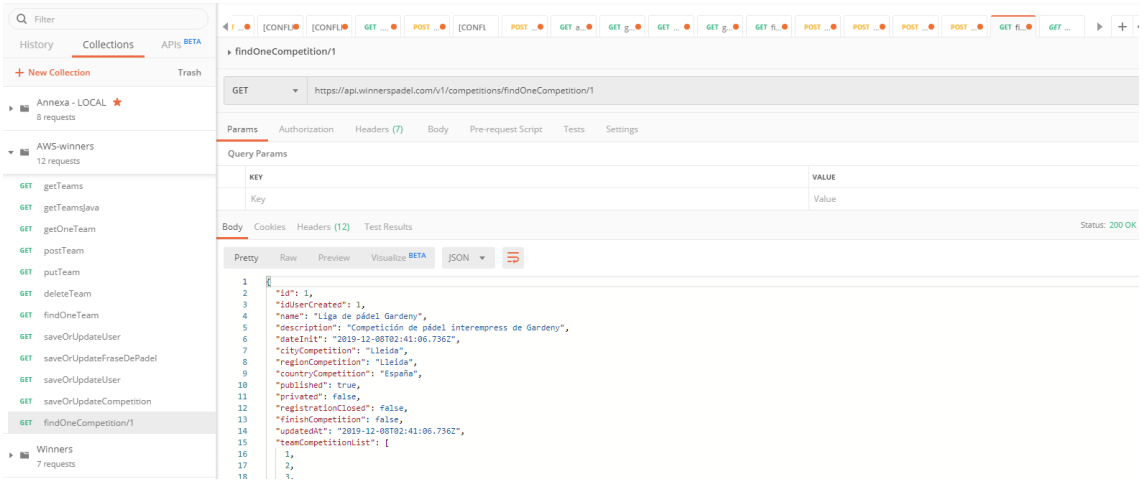
En este caso, se nos ofrece una funcionalidad para realizar llamadas Rest desde AWS, donde podremos validar que a partir de una llamada diseñada, se obtiene una respuesta esperada.



f.36 Resultado de una prueba del API Gateway

Pruebas **“**Postman**”**

Por último, para probar todas las llamadas en formato real, se hizo un proyecto de Postman donde se realizaban llamadas al api de AWS



f.37 Resultado de una prueba Postman

3.2. Recortes en la planificación

Tras estudiar, comprender, desarrollar, corregir, testear, y volver a desarrollar con nuevos conocimientos y estrategias, vi que la planificación inicial no había sido realista, y que era imposible realizar parte de la app con todas las funcionalidades que se habían pensado.

Es por ello que al final he tenido que recortar estas funcionalidades dejando solamente las siguientes:

3.2.1. Login

Se realiza:

- Logueo de usuario contra Cognito
- Validación del estado del usuario según indique Conito

No se realiza:

- Nuevos usuarios:
  - Para la aplicación no es fundamental crear nuevos usuarios, ya que con la existencia de uno se puede validar todo el funcionamiento

- Deslogueo de la app:
  - Cerrando la aplicación se puede salir de app, por lo que no es necesaria la función por el momento

- Securización con JWT
  - Todos los datos de la aplicación se pueden consultar de momento en texto plano si se conoce la forma de llamar a la API. Securizar la aplicación habría llevado aún más tiempo de investigación del bankend para conectar Cognito, ApiGateway y Lamba.
  - Se ha descartado principalmente por el coste de tiempo que habría costado implementarlo

3.2.1.1. Generación y consulta de datos de la competición

Se realiza:

- Consulta de datos de la competición, a nivel de calendario, partidos y clasificación

- Consulta de datos
  - Consulta de información de equipos y jugadores
  - Funciones para ponerse en contacto con estos
  - Consulta de información de los jugadores y sus estadísticas

- Trabajo con los partidos
  - Guardar un resultado, vía ventana de dialogo, y vía marcador
  - Guardar la presencia o no de un jugador a un partido

- Introducción de resultados mediante el modo marcador en el móvil
- Introducción de resultados mediante el modo marcador en el reloj

No se realiza:

- Gestión de usuarios según rol
  - La carga de trabajo habría supuesto la creación de varios usuarios para poder probar con cada uno de ellos los diferentes roles

- Módulo de administración de torneos
  - La aplicación se basa sobre todo en la consulta y guardado de datos de una competición. Generar esta no es la primera necesidad.
  - Se descarta así pues toda la parte de generación de Fases, Rondas, Calendarios y Partidos
  - Todos los datos para simular este método se introducen desde la clase “com.winnerspadel.RellenarBD” que usa como base de datos el archivo “./res/datosBD.json”
  - Es decir, queda descartada toda la interfaz móvil de introducción de datos, pero queda realizado todo el controlador-modelo

- Módulo de notificaciones a usuarios
  - No se realizan las notificaciones en cambios de estado de partidos ya que no son primordiales en una primera versión de la aplicación, y se delega esta responsabilidad de saber cuándo y contra quien se juega, en el propio usuario que podrá consultar los datos entrando en la app de forma más habitual.

- Módulo de marcador en tiempo real
  - Realizar el marcador en tiempo real, habría supuesto estudiar nuevas funcionalidades del backend para entender cómo realizar llamadas a todos los dispositivos conectados que quisiesen seguir un partido en vivo.

3.2.2. Reloj, aplicación wear

Se realiza:

- Marcador de un partido amistoso para que se pueda usar la funcionalidad de contar puntos.
- Iniciar un partido desde el móvil para controlar los puntos de ese partido
- Conexión móvil-reloj para intercambiar el marcador de la aplicación móvil y el reloj.

No se realiza:

- Acceso a los partidos de una competición y carga de estos para controlar los puntos.
    - De momento se tendrá que iniciar esta funcionalidad desde el móvil, lo cual no es un gran problema, por lo que se recorta tiempo desarrollándola.

3.2.3. Explicación de los recortes en la planificación

Se han descartado principalmente todos aquellos apartados que habrían supuesto un nuevo estudio de formas de trabajo y utilización de herramientas.

Estos estudios, y su posterior aplicación suelen llevar siempre más tiempo del planificado, al no conocer el alcance y dimensión de la tarea a afrontar.

Así pues, todo aquello que podría ser factible de generar bloqueos en tiempos de desarrollo se ha descartado si no era totalmente fundamental

Se ha realizado todas las partes importantes de la aplicación, valorando que estas son:

- El backend, ya que, sin datos la consulta de estos y su modificación, no serviría de nada una aplicación de gestión de una liga
- La consulta de datos a nivel de usuario, que es la principal motivación de la realización de este proyecto
- La herramienta para controlar en tipo marcador el tanteo de un partido, que fue la idea primigenia que dio vida a todo el proyecto.
- La misma herramienta de tanteo de puntos, pero en el reloj, dando así el factor diferencial y el toque de personalidad a la aplicación, siendo este una de las partes fundamentales de la app

3.2.4. Principal motivo de la replanificación

El motivo principal de toda la replanificación ha sido la extensión de tiempo que ha tomado realizar todo el backend. Como se comenta en el punto anterior, sin backend no hay aplicación, ya que esta depende principalmente de los datos.

Se han tomado tantas molestias en el backend porque se quería usar, también como factor diferencial del proyecto, toda la tecnología de última generación Serverless ofrecida por Amazon con AWS.

La realización de todo este apartado da entidad y poso al proyecto, pero su desarrollo, comprensión y realización ha sido mucho más costoso de lo inicialmente planificado.

No por ello es menos cierto, que se ha cubierto de forma exitosa todo el desarrollo del backend, y se ha generado una API robusta que permitirá crecer a la aplicación en términos de seguridad, tráfico de datos, y número de usuarios, dando así la posibilidad de que todas las partes que no se han podido desarrollar en esta versión de la aplicación, se pueden hacer en un futuro de forma mucho más rápida y sencilla.

3.3. Cambios en el diseño y explicación de las pantallas

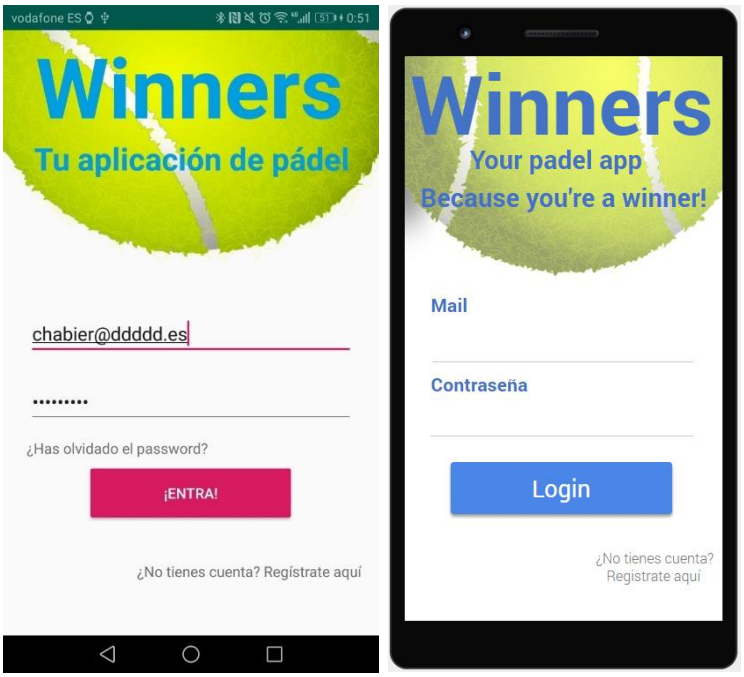
Conforme se ha ido avanzando en el desarrollo se ha podido aprender y valorar nuevas formas de presentar la información, quizá más óptimas y bonitas que las pensadas inicialmente.

En otros casos, este rediseño ha sido una obligación para poder solucionar problemas que ya sea por desconocimiento inicial, o por no haber tenido en cuenta todas las posibilidades, han requerido una adaptación.

Si bien es cierto, que en todo momento se ha respetado el diseño original con su estructura, formas, colores y navegaciones.

3.3.1. Cambios en el login

La pantalla de login se ha respetado completamente, cambiando pequeños detalles.



Por el contrario, debido a los tiempos de espera que supone la consulta asíncrona de los datos al backend, se ha tenido que desarrollar una nueva

pantalla de carga, para que el usuario se sienta más cómodo que esperando en la pantalla de inicio sin poder acceder a nada.

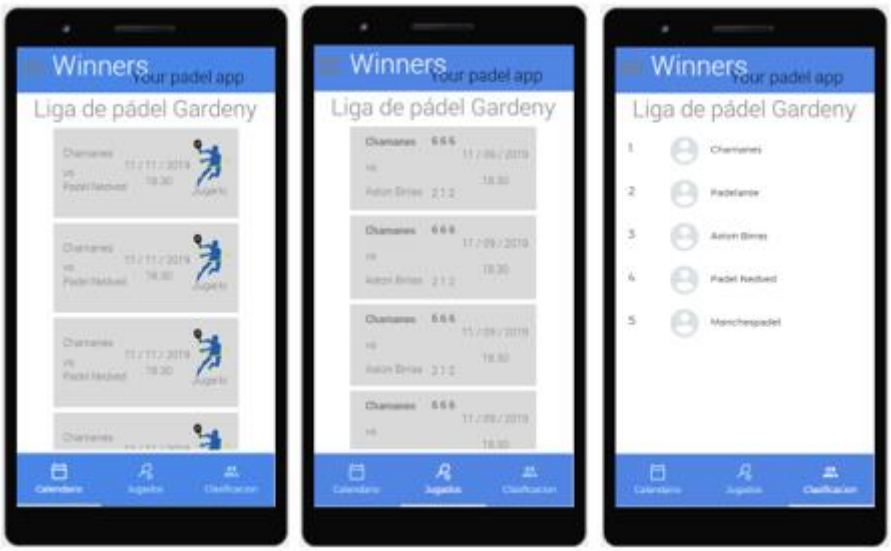
En esta pantalla se muestran 3 ProgressBar de carga que indicarán al usuario el progreso de como se está recuperando la información.

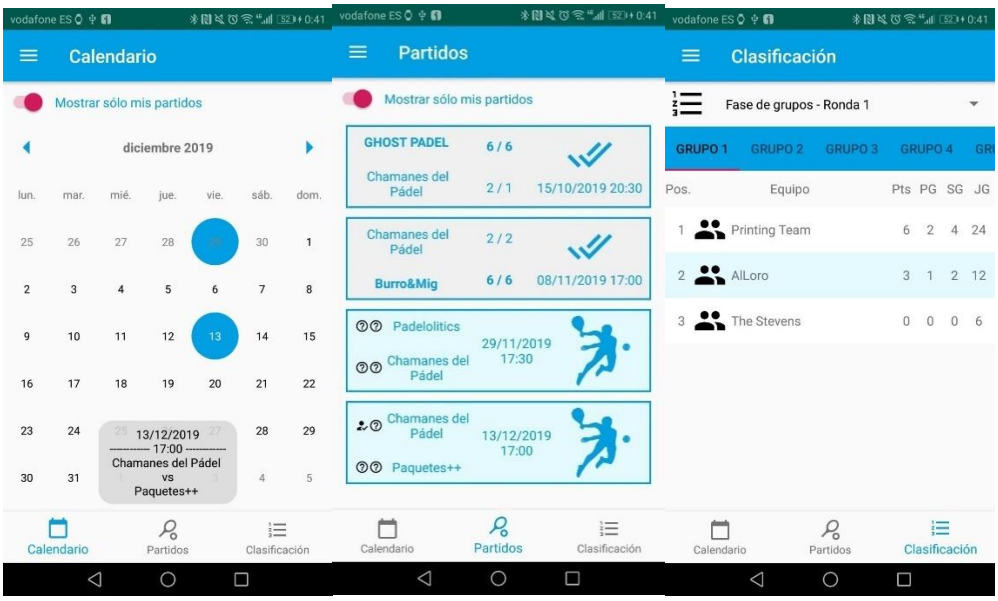
Por otra parte, como licencia creativa, y para hacer más amena la espera al usuario, se ha diseñado un servicio de “Frases de pádel” donde, cada 3 segundos aparecerá una frase recuperada igualmente con llamadas asíncronas al servidor, lo que puede permitir agregar, quitar o modificar frases y que todos los usuarios las tengan actualizadas.

3.3.2. Cambios en la pantalla de inicio

La pantalla de inicio ha respetado la estructura con el menú de botones inferior, así como la idea de qué debía mostrar de información cada una de las opciones del menú, pero es sin duda la que más cambios ha sufrido.

Este cambio, como podemos ver en la siguiente comparativa, ha sido para mejor estéticamente y funcionalmente.





Calendario

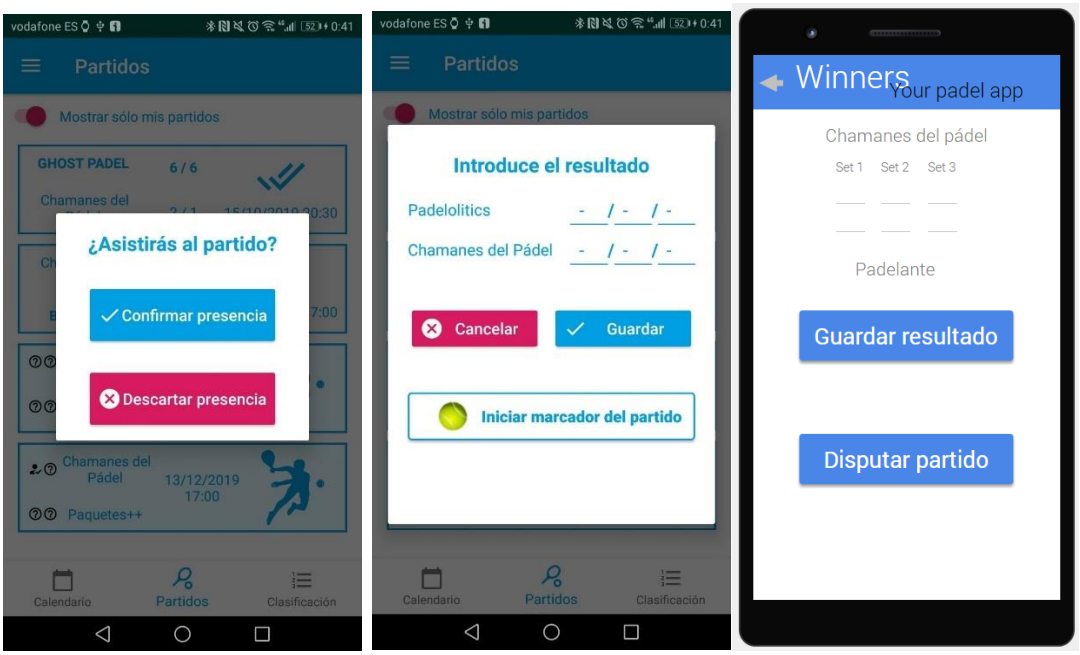
La pantalla de calendario, cobra ahora identidad propia haciendo la función de un calendario.

- Se ha incluido un botón superior para poder ver la información solo del usuario, o de todos los jugadores del campeonato
- Cuando se presione en un día marcado, se mostrarán todos los partidos programados para ese día.

Partidos

La pestaña de “Jugados” también ha adquirido una personalidad aún más fuerte siendo la responsable de mostrar todo los partidos jugados o programados de una forma más explicada que en el calendario.

- Se aprecia el mismo botón superior para ver solo la información propia o la de todos los jugadores
- Se diferencian los partidos jugados pasados, de los partidos futuros programados.
- Se ha incluido unos “checks” de presencia en los partidos futuros donde los jugadores podrán confirmar su conocimiento del partido y si asistirán.
- El usuario solo puede interactuar con sus partidos
    - Se ha incluido la funcionalidad de poder “eliminar un resultado” más pensada hacía los testeos iniciales y las pruebas, ya que el bajo volumen de datos hace que sin estas eliminaciones sea difícil valorar la aplicación. Esta eliminación de datos no es trivial, y no debe mostrarse como se hace ahora, pero se planificará en una próxima versión.



Desde cada partido futuro se mostrará una venta emergente siguiendo la siguiente caustica.

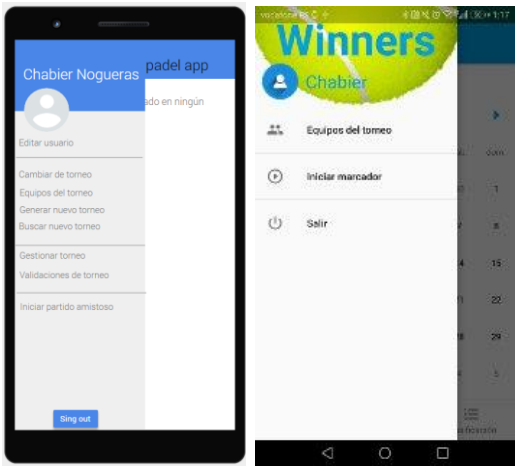
- Si la fecha de disputa de partido es anterior a la actual, se mostrará un dialogo para que el usuario pueda introducir el resultado del partido
- Si la fecha de disputa de partido es posterior a la actual, se mostrará un dialogo para que el usuario pueda confirmar o descartar su presencia en el partido.

Se puede apreciar la mejora estética y funcional de la pantalla definitiva, sobre el diseñada a la hora de introducir el resultado

Clasificación

La pestana de clasificación por su parte, ha sido un caso claro de mala previsión y planificación, ya que no se tuvo en cuenta la estructura de la mayor parte de campeonatos, donde podemos encontrar “Fases” (de grupos y de play off, por ejemplo), “Rondas” (ronda de corte, ronda de clasificación, ronda clasificación 2…), y “Grupos”, donde los equipos se dividen para hacer campeonatos más fáciles de manejar (primera división, segunda división)

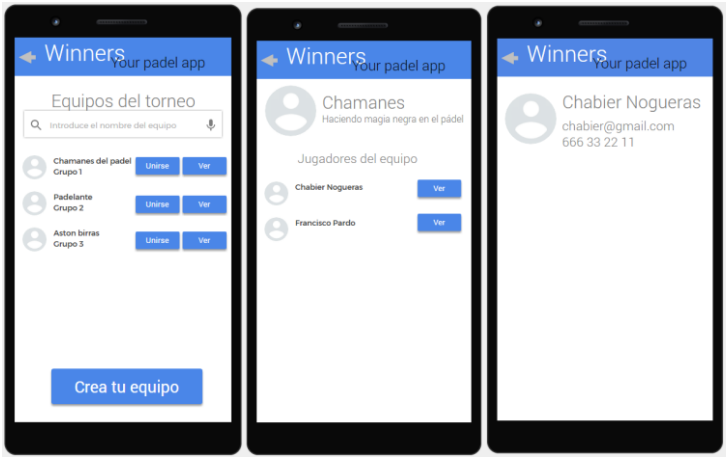
3.3.3. Cambios en el menú

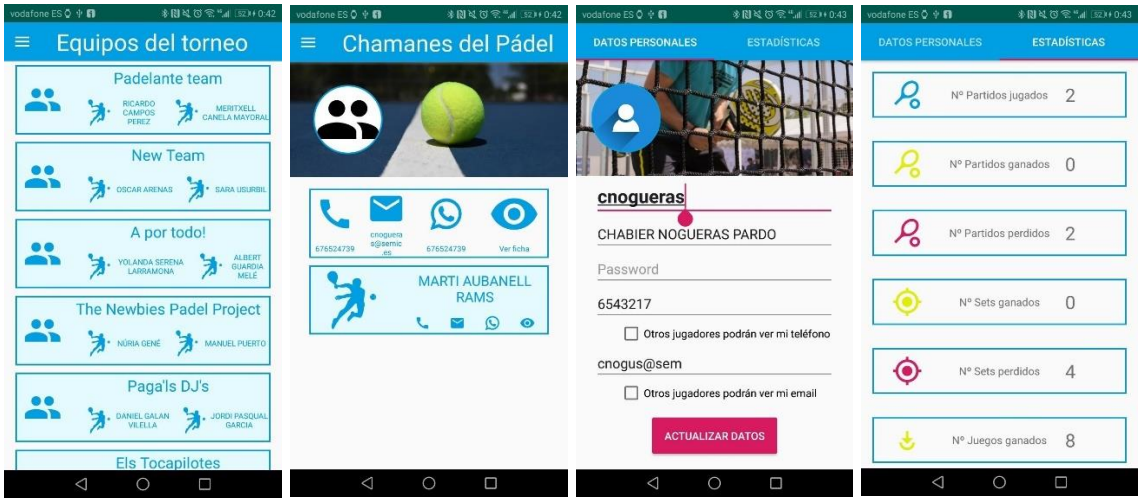


Se ha respetado el diseño de menú lateral, pero se ha mejorado la estética siguiendo las líneas de la aplicación, y se han reducido el número de opciones del menú que se han podido afrontar.

Pon contra, desde la cabecera, se ha agregado la funcionalidad de que se pueda acceder a la información del usuario logado.

3.3.4. Equipos del torneo y jugadores

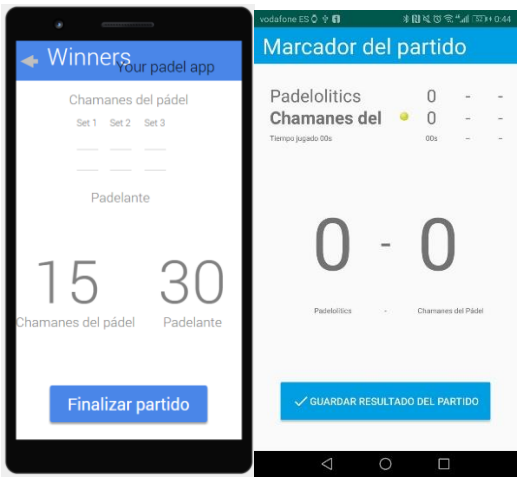




La mejora estética es evidente en las pantallas diseñadas para mostrar la información de los usuarios y sus equipos, pero se ha respetado en la medida de lo posible el diseño e idea inicial

- El listado de equipos es más manejable y fácil de leer.
- La pantalla del equipo muestra un banner descriptivo superior que será editable en el futuro, junto con el escudo del equipo.
- Se ha agregado una funcionalidad interesante:
    - Cuando se presiona encima de un jugador, la ficha se da la vuelta, y muestra 4 accesos directos para llamar al jugador, enviarle un correo electrónico, enviarle un whatsapp aunque no lo tengas guardado en tu agenda, y poder acceder a la ficha de la app de él.
- En la ficha del jugador, se muestra un banner superior que junto con la imagen del jugador, será editable en el futuro.
    - La pantalla mostrada aquí es la de la “edición del usuario de la aplicación” ya que es la misma, pero con campos editables, y se sintetiza para no repetir información.
    - Información estadística del jugador, todos los partidos que ha disputado, así como los resultados obtenidos en ellos a modo de resumen.

3.3.5. Marcador de los partidos



Esta pantalla se ha respetado en su diseño, aunque se ha mejorado algo la estética.

- El botón de “guardar el resultado” solo aparece cuando se ha iniciado el conteo del marcado desde un partido programado.
- Desde un partido amistoso, no aparece el botón, y los nombres de los equipos son “local y visitante”

3.4. Frontend: App Android

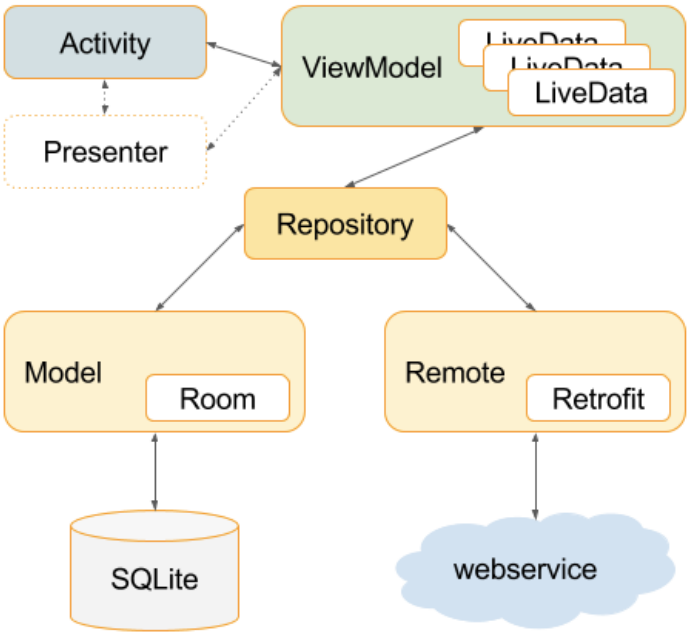
En el siguiente apartado se explicarán las decisiones de diseño de la arquitectura de la aplicación, así como los problemas encontrados y la solución adoptada.

3.4.1. Arquitectura CLEAN

Desde hace relativamente poco tiempo, se está empezando a usar una arquitectura de nominada CLEAN, consistente en los siguientes principios [12] [13] y [15]

- Single Responsibility: Cada unidad de código debe tener una única responsabilidad.
- Open/Closed: Podemos extender las funcionalidades de un objeto, no modificarlas.
- Liskov: Un objeto puede ser siempre sustituido por otro objeto subtipo
- Interface segregation: Mejor varios interfaces específicos que uno general.
- Dependency inversion: Un objeto debe depender de abstracciones.

En la medida de lo posible, se ha intentado seguir esta arquitectura en el desarrollo de la aplicación.



f.38 Estructura de una aplicación CLEAN

Así pues, se han usado los siguientes puntos

3.4.1.1. ViewModel

Se recomienda separar por capas la actividad de presentación de datos y su tratamiento. Para ello se usa ViewModel, donde se podrán tratar los datos que se muestran.

3.4.1.2. LiveData [16]

Está estrechamente relacionado con ViewModel, ya que gracias a estos nuevos objetos, se pueden mostrar los cambios en un objeto de forma asíncrona.

El LiveData se ha usado de forma amplia en la aplicación para mostrar los datos con forme se iban obteniendo y cargando con las llamadas asíncronas al servicio web.

El mayor uso ha sido en la pantalla de inicio, aunque alguno de estos livedata se ha enviado a la pantalla de “loading” para mejorar la experiencia de uso del usuario.

3.4.1.3. Retrofit [17]

En este caso, se ha producido una larga historia de hechos que han supuesto muchos retrasos en la aplicación, y que se pueden resumir como el desconocimiento de la existencia de esta librería.

Inicialmente se pensó en generar una librería propia de consulta al webservice, de forma que esta librería se pudiese usar en otras futuras aplicaciones, o una web.

Una vez desarrollada, lo cual no fue trivial, para incluir esta librería se pasó por diferentes fases, todas ellas fallidas.

- Se intentó incluir como un “jar”, dando problemas con dependencias.
- Se intentó incluir como un proyecto de Jitpack [14] haciendo una dependencia a partir de un repositorio, pero nuevamente fue en vano.
- Se incluyó directamente como un módulo del proyecto, pero nuevamente muchas clases chocaron.
- Por último, se intentó incluir todo integrado dentro de la propia aplicación, y así se consiguió usar la librería propia construida en base a peticiones HTTP

Tras toda esa inversión de tiempo y esfuerzo, se descubrió “retrofit” [17], una librería capaz de hacer llamadas REST y tratar las respuestas con poco más que indicar la url y el objeto a enviar o recibir.

Todo el tiempo que se perdió hasta encontrar esta solución, posteriormente se ha ganado al ser extremadamente fácil el uso de esta librería para hacer llamadas REST.

3.4.1.4. Model Room (Cache) [21]

Otro de los problemas encontrados fue como almacenar la información y como poder acceder a ella desde diferentes partes de la aplicación.

Inicialmente no se pensó en como almacenarla, ya que se esperaba recuperar la información vía REST cada vez que se necesitase.

Sin embargo, la experiencia de uso y programación desaconsejó esta actuación, por lo que se tuvo que reaccionar buscando la mejor manera posible para almacenar y consultar los datos.

Se pensó en utilizar una base de datos SQLite, pero teniendo en cuenta que los datos que se usan vienen en formato Json desde una base de datos NoSQL, y que este formato es óptimo para mejorar en velocidad de consulta, así como de mostrado de datos, tener que tratar estos para almacenarlos correctamente en tablas en el móvil no pareció la mejor opción.

Las SharePreferences se valoraron también, pero la funcionalidad de estas es la de almacenar configuraciones de la app, y no los datos de esta.

Por último y de forma definitiva, se apostó por guardar los datos temporales recogidos desde la llamada REST al Webservice, en la estructura que está pensada para almacenar datos temporales: La cache.

Así pues, los datos json de la competición, usuarios, equipos, partidos… se almacena en la memoria cache, siendo esta eliminada cuando se accede a la aplicación para forzar a tener siempre los últimos datos.

3.4.2. Librerías importantes usadas

3.4.2.1. WinnersPadelCoreLIB

Como se ha explicado en la introducción, este proyecto nace de una necesidad, la de controlar una liga de pádel que hace más de un lustro que se disputa.

En ese tiempo ha habido diferentes intentos de desarrollo de soluciones, con mayor o menor éxito.

Uno de los casos de existo fue la de la librería “WinnersPadelCoreLIB” que permite llevar el conteo de puntos en un partido de pádel, y almacenar cuando se introducen estos para poder después generar estadísticas del partido.

La librería es liviana, por lo que se ha incluido tanto en la aplicación móvil con en el reloj para las pantallas de marcador.

Esta librería, desarrollada también por mí, ha hecho que se ganase tiempo en el desarrollo de las funcionalidades de marcador.

También, como almacena la información de los partidos, sirvió de base para desarrollar la estructura básica de la Base de Datos.

3.4.2.2. Material-CalendarView [18]

El módulo de calendario que ofrece Android para usar, es sencillo, y con pocas funcionalidades, pero a la vez es lo suficientemente abierto como para que terceros hayan desarrollado módulos de calendarios que lo mejoran.

Después de probar muchas de estas librerías, todas con fallos más o menos importantes, como no poder cambiar a lunes el día de inicio de semana, o que se perdiesen las referencias de las marcas al cambiar de mes, se decidió por usar la librería “Material-CalendarView”, que, aunque es simple, ofrece una funcionalidad con todas las características buscadas, y no tiene fallos evidentes.

Sus principales inconvenientes son:

- No indica de forma diferente el día actual
- No permite marcar en el calendario con diferentes colores

Pero ambos dos son problemas salvables.

3.4.2.3. Lombok [19]

Más que una librería que aporte funcionalidad a la aplicación, es una librería que aporta velocidad de desarrollo y elimina código “BoilerPlate”, es decir, repetitivo

Gracias a anotaciones como @Data, @AllArgsConstructor, @NoArgsConstructor o @ToString se consigue eliminar código visual (aunque sí se genera en tiempo de compilación) y se ahorra muchísimo tiempo de desarrollo.

3.4.2.4. Jackson y Gson [20]

Ambas librerías tienen el mismo objetivo, trabajar con datos Json para poder navegarlos, o serializarlos o deserializaros como objetos Java.

Es por ello que, teniendo en cuenta que toda la capa de datos de la aplicación está en Json, son librerías correctas para usar.

Se ha utilizado más Jackson, ya que está más actualizada, pero da ciertos fallos con ciertas estructuras, que ha hecho que se tenga que usar Gson para tratar otra información sin fallos.

3.4.2.5. AWS - Cognito

Para realizar todas las consultas al servicio de Cogito se Amazon, y recuperar la información y el token de uso, se han incluido y usado estas librerías.

3.5. Modulo wear

Este módulo, aunque pequeño, tiene una gran importancia, ya que es el factor diferencial por el que apuesta la aplicación.

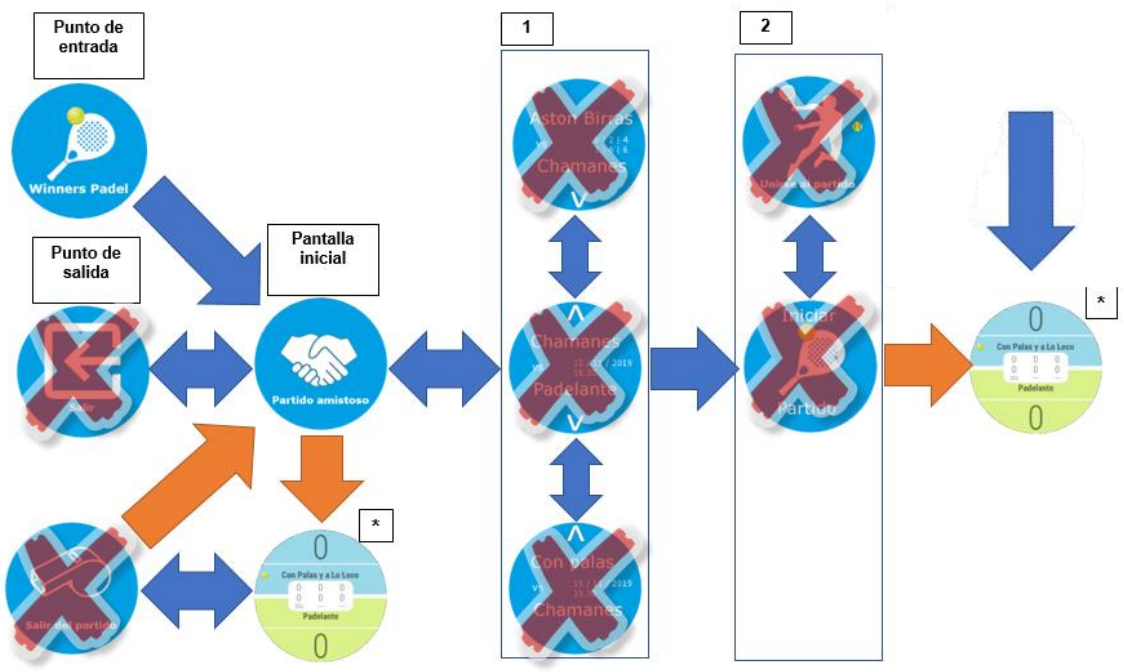
Como se explicó en la presentación, uno de los problemas a la hora de disputar un partido de pádel amateur, es el perder la cuenta de los puntos.

El módulo del reloj es importante en este aspecto ya que permite a los usuarios que dispongan de uno de poder llevar la puntuación en tiempo real y sin perder la cuenta.

Es por ello que era importante que se desarrollase, aunque fuese de forma mínima.

3.5.1. Recortes en la planificación

Debido a los retrasos anteriormente explicados y justificados, este módulo también quedó afectado, pasando a tener la siguiente perspectiva



f.39 Rediseño de la funcionalidad del reloj

Se ha desarrollado el módulo principal de marcador, y se han implementado 2 formas de llegar a él.

- Desde el inicio de la aplicación, arrancándola en modo partido amistoso

- Desde la aplicación móvil, arrancándola en modo marcador de partido oficial.

3.5.2. Rediseño de las pantallas

En este caso, en las 2 pantallas que se han preparado, se ha respetado el diseño original, más allá de alguna variación en los colores.

f.40 Rediseño de las pantallas del reloj

3.5.3. Conexión bluetooth con el reloj

La librería “WinnersPadelCoreLIB” desarrollada anteriormente a este proyecto, estuvo pensada para resolver una necesidad que en su momento fue el tanteo de puntos.

En ese instante se empezó a investigar sobre el desarrollo de la programación ellos dispositivos “wareables”, lo que en cierta medida dio pie de formo primigenia a este proyecto.

En ese instante se investigaron apis de Google para poder realizar la conexión bluetooth entre móvil y reloj, y se adoptaron algunas de las soluciones propuestas.

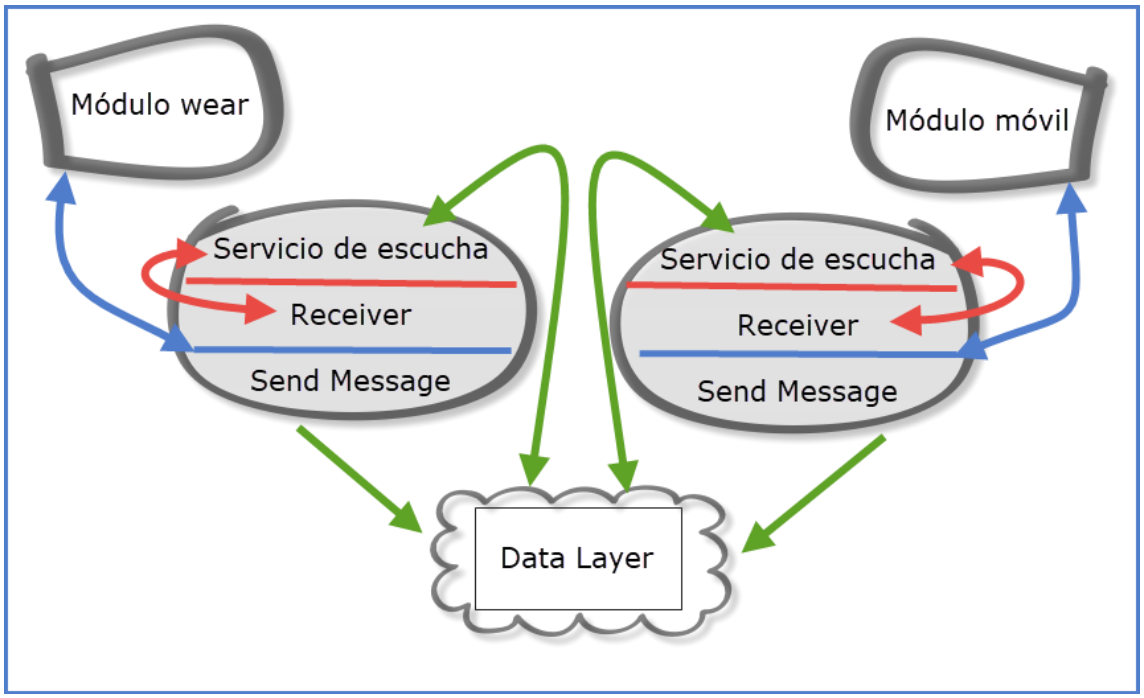
Sin embargo, a día de hoy, esas soluciones han quedado obsoletas y han tenido que ser descartadas para la realización del proyecto, es por ello que se ha vuelto a realizar otra investigación

De entre todas las webs de consulta de información consultadas, ha sido especialmente mencionable esta [21].

La conexión entre móvil y reloj se usa utilizando la capa “Data Layer” o de intercambio de información.

Para acceder a ella, hay que iniciar servicios en el móvil y reloj, y definirlos en el manifest de ambos módulos.

Estos servicios deberán compartir una ruta de intercambio de mensajes, la cual se ha llamado “/WinnersPadelScore”



f.41 Estructura de la conexión entre móvil y reloj a través del Data Layer

A este servicio se le da la posibilidad de realizar diferentes acciones de intercambio de información y se le asocia con la clase que controlará las llamadas a esta capa

```
<service
    android:name=".conexionMobile.MessageService"
    android:enabled="true"
    android:exported="true" >

    <intent-filter>
        <action android:name="com.google.android.gms.wearable.DATA_CHANGED" />
        <action android:name="com.google.android.gms.wearable.MESSAGE_RECEIVED" />
        <action android:name="com.google.android.gms.wearable.CAPABILITY_CHANGED" />
        <action android:name="com.google.android.gms.wearable.CHANNEL_EVENT" />
        <category android:name="android.intent.category.LAUNCHER" />
        <data android:scheme="wear" android:host="*" android:pathPrefix="/WinnersPadelScore" />
    </intent-filter>
</service>
```

f.42 Trozo del manifest de la app de reloj donde se muestra el filtro que escuchará el Data Layer

La clase que actuará como servicio de escucha, llamada “MessageService” tiene que extender de “WearableListenerService”

Cuando esta clase, a nivel de aplicación, recibe un mensaje, mediante una llamada de tipo “LocalBroadcastManager” extenderá este a todos los “Receiver”.

Los Receiver extienden de “BroadcastReceiver” y trabajan a nivel de módulo. Al recibir un mensaje, tratan la información para mostrarla en el dispositivo.

Para enviar un mensaje, se ha implementado una clase que actuará como un hilo independiente para, de forma asíncrona, no bloquear la aplicación mientras se envía la información.

En esta clase se consultan los nodos que están en la misma red, es decir, todos los módulos asociados al mismo proyecto, y se envía un mensaje específicamente a ese, que será escuchado por el “WearebleListenerService”

```
public void run() {

    Task<List<Node>> wearableList = Wearable.getNodeClient(context).getConnectedNodes();
    try {

        List<Node> nodes = Tasks.await(wearableList);
        for (Node node : nodes) {
            Task<Integer> sendMessageTask = Wearable.getMessageClient(context)
                    .sendMessage(node.getId(), path, message.getBytes());
        }

    } catch (ExecutionException | InterruptedException exception) {
        exception.printStackTrace();
    }
}
```

f.43 Código que escucha el Data Layer

3.5.4. Cuando se inicia la aplicación

Debido a los recortes para poder llegar a la entrega, la aplicación sólo se puede arrancar de tres formas diferentes.

- Desde la propia aplicación del reloj, en modo partido amistoso
- Desde la aplicación del móvil, en modo de partido amistoso
- Desde la aplicación del móvil, para controlar la puntuación de un partido oficial

Al iniciar el marcador desde el móvil, la aplicación del reloj detectará este cambio y auto iniciará la aplicación del reloj.

3.5.5. Cómo desplegar y probar una app de reloj

En este caso nos encontramos con 3 casuísticas.

Cuando la aplicación esté subida a Google Play, y esta se instale, se desplegará el módulo wear en el reloj y se podrá probar.

*Nota: mi ordenador no dispone del clip en el procesador que permite virtualizar sistemas AVD para iniciar las aplicaciones en el propio equipo, es por ello que me he visto obligado a usar mi propio móvil y reloj para probar de forma real las aplicaciones.

A nivel de debug, que es donde nos encontramos, para desplegar e iniciar la aplicación en el reloj, deberemos seguir los siguientes pasos.

Google lo explica de manera oficial [24] pero hay pasos en los que es importante hacer hincapié

Activar las opciones de desarrollo en el reloj:

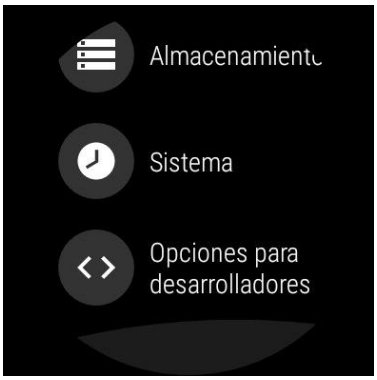
Acceder a:

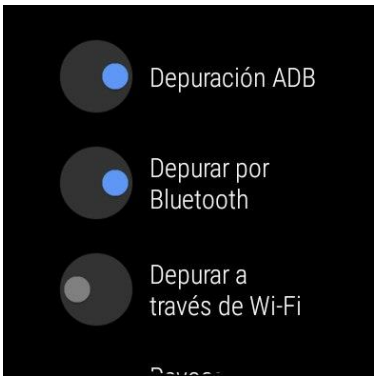
- Ajustes -> Sistema -> Información -> presionar varias veces seguidas en Número de compilación

Activar la depuración ADB en el reloj:

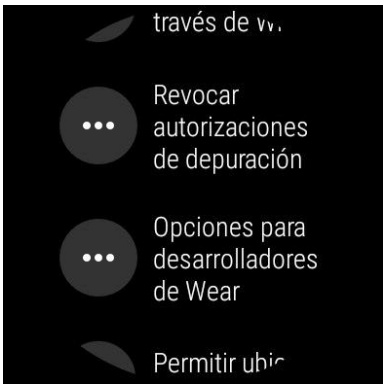
Acceder a:

- Ajustes -> Opciones de desarrolladores (nuevo menú activado anteriormente)
    - Encender la depuración por ADB
    - Encender la depuración por bluetooth
    - O encender la depuración por wifi si se dispone de ella





Puede pasar que a veces el ordenador de desarrollo no termine de conectarse al reloj, para ello hay que, en el mismo menú, deberemos “Revocar autorizaciones de depuración” para que de nuevo se vuelvan a pedir.



Si vamos a depurar la aplicación por Wifi, tan solo nos quedaría conectarnos al reloj desde el ordenador como indicaré a continuación.

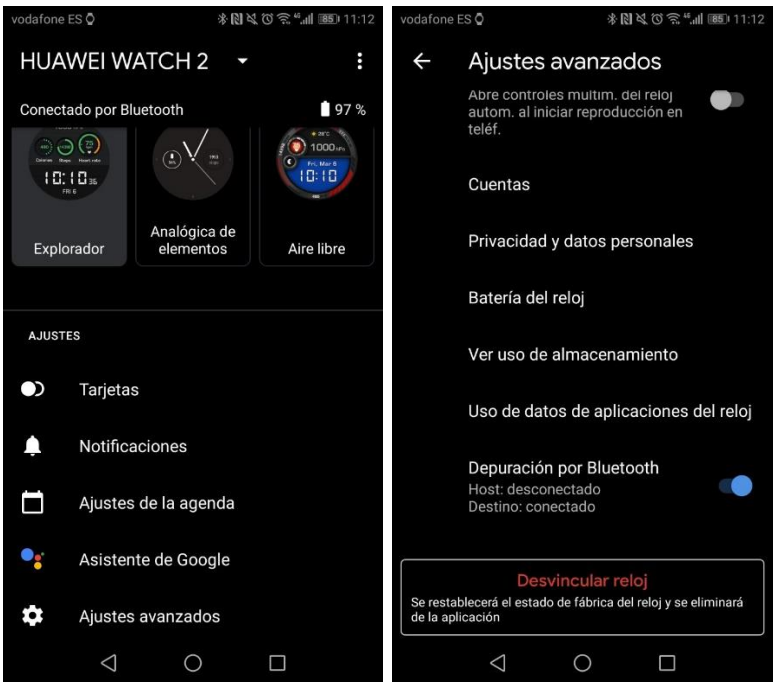
Pero si vamos a hacerlo vía Bluetooth, aún tendremos que realizar otro paso:

Una vez activada la depuración en el reloj, deberemos acceder a la aplicación “Wear OS” de Google para controlar el reloj desde el móvil.

Allí bajaremos hasta los ajustes avanzados, y en la pantalla que se nos abrirá activaremos la opción “Depuración por Bluetooth”.

En ese instante deberíamos ver como “Destino” cambia a “conectado”.

Faltaría así pues solo la parte de conectar el móvil al equipo y conectar host también



Para terminar la conexión deberemos conectar el móvil al ordenador, y con las opciones de desarrollo activadas en el móvil, acceder a Android Studio.

En él, una vez que ya tengamos sincronizados Android Studio y el móvil, deberemos abrir el terminal y ejecutar los dos siguientes comandos de ADB.

Si no se dispone de ADB (la herramienta de Google para debugar dispositivos wearables, entre otras funciones) se puede acceder a un artículo que comenta como instalarlo [22], instalar desde aquí, donde encontraremos el enlace al archivo zip en Google Drive: ADB Installer v1.4.3.zip

En el terminar propio que ofrece Android Studio (o en otro que se desee) se deberán ejecutar los siguiente comandos:

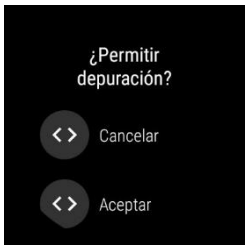
- Para iniciar la depuración por wifi
  - adb connect 192.168.1.100 (siendo la dirección IP la del reloj)

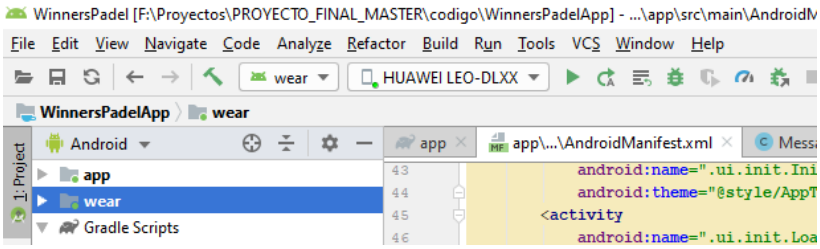
- Para iniciar la depuración por bluetooth
  - (se recomienda revocar los permisos en el reloj en este instante si los siguientes comandos no funcionan)
  - adb forward tcp:4444 localabstract:/adb-hub
  - adb connect 127.0.0.1:4444

```
F:\Proyectos\PROYECTO_FINAL_MASTER\codigo\WinnersPadelApp>adb forward tcp:4444 localabstract:/adb-hub

F:\Proyectos\PROYECTO_FINAL_MASTER\codigo\WinnersPadelApp>adb connect 127.0.0.1:4444
connected to 127.0.0.1:4444
```

A continuación, el reloj nos pedirá permiso para proceder a la depuración.





Ya solo nos quedará desplegar el módulo “wear” en el reloj desde Android Studio.

3.5.6. Pruebas de funcionamiento

Como ya se ha comentado, la principal prueba de funcionamiento es el testeo de la librería para contar los puntos.

Esta librería se ha testeado en los 2 últimos años en múltiples partidos, dando un resultado correcto, si bien es cierto que podría existir algún punto de mejora.

3.6. Problemas encontrados (resumen)

3.6.1. AWS, nueva tecnología

En una aplicación de gestión de información, el backend toma una importancia vital, pues es el origen de los datos a tratar.

Es por ello que se ha tratado que este sea robusto y fiable, así como seguro.
La mejor opción en la actualidad es sin duda alguna utilizar los servicios Serverless, y en ese apartado hoy en día Amazon es líder indiscutible con su sistema AWS.

Desgraciadamente, afrontar este desarrollo sin conocer bien la tecnología ha supuesto una inversión de tiempo mucho mayor de la esperada, aunque los resultados obtenidos han sido sensiblemente mejores a los también esperados.

Y ya no solo la tecnología, si no que hacer un buen diseño de la arquitectura para trabajar con los datos el vital si se quiere afrontar un desarrollo al nivel de exigencia establecido en este proyecto. Los cambios en la arquitectura han supuesto programar y reprogramar varias veces el backend

3.6.2. Bloqueos en la programación

Trabajar con librerías desconocidas, o intentar integrar las de propio desarrollo puede suponer en algunos casos bloqueos que no permiten avanzar en el desarrollo, y que pueden llegar a ser causados por terceros. Buscar solución a estos problemas a veces no es tarea fácil.

Android está pensando para trabajar con estados, no solo en el ciclo de vida de las activities, sino también en otros muchos objetos.

Uno de los que más afectó a los tiempos de programación fue el TabAdapter.

Con esta herramienta se pueden hacer pestañas en la aplicación, sin embargo, requiere de un tratamiento especial conforme a su estado.

Al iniciarse el objeto que implementa TabAdapter, instancia todos los Fragments que va a usar en las pestañas, siendo más precisos, exactamente instancia el fragment del tab 1 y 2, y cuando pasas al 2, carga el de 3, etc.

Si sales de la pantalla de los tabs y vuelves a entrar con otros datos, el TabAdapter mira primero si ya tiene los fragments en el FragmentManager, y si los tiene no los vuelve a instanciar no mostrando así la información real.

La solución, es cambiar el tipo de herencia del Adapter para decirle que extienda de FragmentStatePagerAdapter, el cual siempre recarga los datos de los fragmens.

Este bloqueo, tan simple como importante, nos hizo invertir mucho tiempo para poder:

- mostrar los datos de la clasificación, donde existía un tab dependiente de un spinner
- mostrar los datos de usuario y sus estadísticas
- mostrar los jugadores de un equipo pudiendo acceder a sus datos

3.6.3. Tiempo de carga pidiendo los datos a AWS

Uno de los problemas no planificados que hubo que afrontar fue la baja latencia en los tiempos de carga desde AWS.

Como se ha comentado, el funcionamiento del serverless mediante lambda consiste en entregar el código a AWS y dejar que sea él el que decida cuando levantar la aplicación.

Al ser una aplicación java, los tiempos de inicio son aún mayores ya que se requiere más capacidad de proceso.

Esto afectó al rendimiento de la aplicación, por lo que hubo que buscar una solución, la cual fue cargar todos los datos al iniciar la aplicación en, la pantalla de loading.

Para hacer menos larga la espera se agregó la muestra de frases de pádel graciosas.

3.6.4. CalendarView, que librería usar

Este punto, ya comentado anteriormente, también provocó retrasos, ya que se tuvo que investigar y probar una a una cada una de las librerías que se iban encontrando para realizar esta función.

3.6.5. LiveData, cuando usarlo

LiveData, herramienta englobada en los principios Clean, es una buena forma de trabajar, sin embargo, requiere del conocimiento de cuando aplicar consultas síncronas (SET) o asíncronas (POST), como hacer estas llamadas, y como tratar la información devuelta.

Siendo una herramienta potente, tiene, como todo, la dificultad de entenderla como novedad que es, y saber aplicarle un correcto funcionamiento.

3.6.6. Diseño de la app, como hacerla atractiva

Un punto también importante, y que quizá no se le otorgó tanta relevancia como cabría esperar, es el del diseño visual de la aplicación.

Gracias a Justinmind se pudo realizar unos trazos bastante cercanos a lo que se esperaba diseñar, pero a la hora de programar, en unos casos se puede hacer que determinadas partes queden visualmente mejor o sean más bonitas, y otras hay que hacerlas de otra forma ya que la tecnología lo limita así (de una forma afrontable)

Todo esto ha hecho que en múltiples ocasiones, aunque ya se sabía la base de cómo iba a ser la pantalla, se tuviese que invertir mucho tiempo en hacer retoques en posiciones, colores, iconos y estructuras.

3.6.7. Afectación al calendario de planificación

Todos los puntos detallados, han afectado al calendario de la planificación.

A nivel visual se puede observar en el diagrama de Gantt el pequeño desbarajuste organizativo, ya que se ha tenido que volver atrás en varias ocasiones.

A nivel temporal, ha requerido pensar y decidir recortes en la planificación para priorizar las partes más importantes y que sí o sí debían de estar terminadas, como son

- Backend con serverless
- Login con conito
- Módulo de consulta de partidos a nivel usuario
    - Información del calendario, de los partidos y la clasificación
    - Con la interacción de resultados y presencia.
- Módulo de consulta de información de participantes a nivel usuario
- Módulo de wear para poder tener el marcador en el reloj

- Con inicio de partidos amistosos desde el móvil o reloj
- Con inicio de partidos oficiales desde el móvil

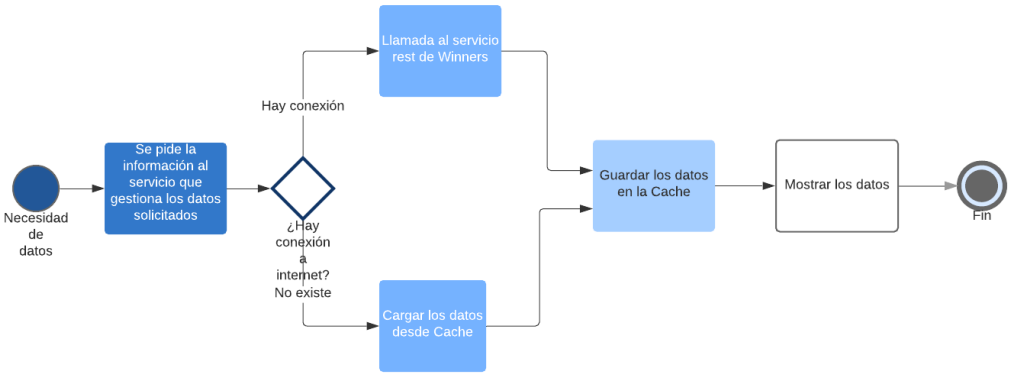
Se deja para una siguiente fase de desarrollo el resto de módulos planificados.

Así pues, el día 20 de noviembre, a mitad exacta de desarrollo y previendo que no se iban a cumplir los plazos, se consultó con el tutor la replanificación explicada para poder afrontar los desarrollos que quedaban.

3.7. Diagramas de funciones destacadas

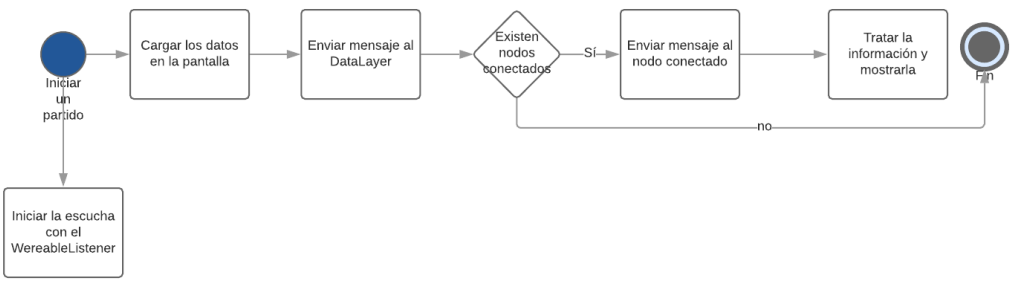
Una vez realizada la aplicación, se ha considerado incluir los siguientes diagramas explicativos de las varias características que requerían su interpretación.

3.7.1. Recuperar datos Rest/Cache



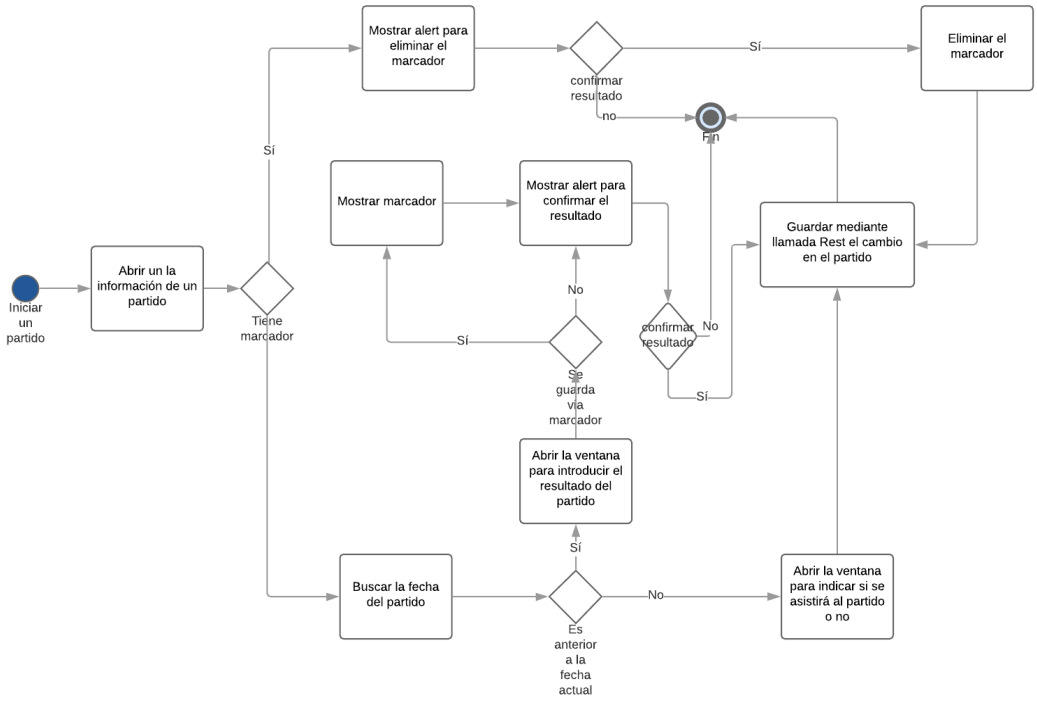
f.44 Diagrama de decisión: Recuperar datos Rest/Cache

3.7.2. Intercambio de datos móvil-reloj



f.45 Diagrama de decisión: Intercambio de datos móvil-reloj

3.7.3. Introducción de datos de un partido



f.46 Diagrama de decisión: Introducción de datos de un partido

3.8. Video-instrucciones y demostración de uso en móvil y reloj.

Aunque ya se ha explicado el funcionamiento de la aplicación en esta memoria, se ha decidido realizar un video explicativo de todas las funcionalidades de las que esta dispone.

f.47 Figura representativa del video explicativo de youtube con las instrucciones

https://youtu.be/QeKJ9FyM4tM

4. Conclusiones

4.1. ¿Qué lecciones se han aprendido del trabajo?

Realizar una aplicación, por sencilla que pueda parecer en los planteamientos iniciales, requiere de un esfuerzo importante que, si no está bien planificado, organizado y estructurado, puede llevar a no concluir el desarrollo.

Así pues, la realización de un correcto análisis, invirtiendo el tiempo necesario que corresponda, con un estudio de mercado para adaptar el producto a las necesidades reales, e investigar competidores para encontrar el nicho de mercado, es una pieza clave en el inicio de un proyecto si se quiere que este sea exitoso.

Es decir, por tentador que pueda parecer, no se puede iniciar un proyecto que se espera exitoso, saltándose la fase inicial de consolidación de ideas.

Por otra parte, la estimación de tiempos y cargas de trabajo sigue siendo una de las mayores complicaciones a la hora de realizar un proyecto. Habría que usar herramientas y conocimientos de expertos para poder realizar un cálculo aproximado y también real. En nuestro caso, se han cumplido plazos, pero cargando de horas extra toda la planificación. De las 388 horas planificadas, se ha pasado a 454 horas. Un total de 66 horas de más.

Se ha aprendido también a realizar un back-end firme que pueda asumir la carga de trabajo, lo cual era toda una incógnita antes de desarrollar todo el proyecto. Específicamente, se ha aprendido a usar el concepto serverless que será común en el mercado en los próximos meses y años.

Este aprendizaje del back-end da como consecuencia la lección de no tener miedo a afrontar retos que inicialmente parecen complicados, ya que con una correcta planificación, se puede asumir el aprendizaje y la utilización de nuevas tecnologías que al final hacen más fácil el desarrollo.

4.2. Reflexión crítica de los objetivos planteados

Cabría hacerse, de mejor manera, la pregunta de ¿Cuáles eran realmente los objetivos planteados?

En un plano más simple, podríamos decir que los objetivos se han cumplido a medias, ya que para el 3 de enero se esperaba tener una aplicación completamente funcional, y todos los obstáculos encontrados han hecho imposible completar todas las funcionalidades, aunque sí es cierto que la aplicación es usable y totalmente estable.

En un plano más elaborado, podríamos decir que los objetivos se han cubierto por completo, ya que con el proyecto se buscaba conseguir un objetivo: realizar una aplicación desde 0, planificando el inicio, documentando el proceso, y presentando todo el conjunto de una forma finalizada. Y en este plano se ha conseguido de forma exitosa.

Como resumen, la mejor respuesta es que, si bien no se ha conseguido desarrollar por completo la aplicación, sí se ha sabido hacer frente a los problemas encontrados, reorientar cambios, adaptarse a los estos y producir una aplicación final, que es la demostración de todo un proceso de trabajo duro, exigente y valioso, que si bien no culmina con una aplicación desarrollada como se proyectó en un principio, sí que culmina con un aprendizaje que permitirá desarrollar aún más el proyecto inicial, haciendo este más potente.

Es decir, en una reflexión final con un mejor punto de vista, se puede asegurar que sí se han cumplido los objetivos planeados, ya que, más que cumplir una idea al milímetro, el objetivo era aprender a afrontar un desarrollo completando una aplicación funcional.

4.3. Análisis crítico del seguimiento de la planificación

Todo proyecto parte de una idea, que conforme se va plasmando en diseños, investigaciones, desarrollos y modelados, va tomando forma.

Realizar una planificación es una tarea difícil, que tiene que prever muchos obstáculos, alguno de los cuales es totalmente imprevisible. Es por ello que, si analizamos la planificación de una forma general, buscando las fechas de entrega como puntos clave, esta sí se ha seguido y se han cumplido los plazos.

Ahora bien, si entramos al detalle de cada tarea, cada diseño, cada pequeña meta planificada, podemos decir que esta planificación no ha sido más que un guion que ha servido de base para conocer el camino y poder ir andando con un objetivo y unas metas, mientras se iban adaptando los pasos a las piedras encontradas.

Grosso modo, se ha seguido la planificación, en cuestión de fechas clave y conceptos de desarrollo en cada una de las fases.

Por el contrario, la planificación detallada del trabajo, ha valido para conocer qué pasos había que hacer y formar así una metodología que, como ha demostrado la presentación final, ha sido correcta. Es decir, aunque al detalle no se hayan cumplido los días y horas de cada tarea, sí se ha cumplido el espíritu de estas, permitiendo así conseguir objetivos.

Todo esto no quita que, haya que haber parado a mitad de desarrollo y se haya tenido que revisar la situación. Tarea que, se cumplan los plazos o no, siempre se debería realizar, ya que el desarrollo de un proyecto conlleva a veces cambios inesperados que hay que saber que existen, que van a aparecer y que hay que saber afrontar.

Debido a esto, aunque es difícil planificar una replanificación, había que contar con ella, como así fue. Gracias a esta replanificación, se consiguió reorientar el camino y plasmar de forma más real los objetivos, haciendo así posible su correcta culminación.

Podríamos concluir así que, en la planificación inicial, faltó planificar la replanificación.

4.4. Futuro de Winners Padel

La replanificación del proyecto ha dejado en el aire ciertas partes que se deberán desarrollar en un futuro cercano si se quiere llevar a una mayor cantidad de usuarios la aplicación.

Actualmente, ésta es funcional para el caso de uso actual, de una liga determinada, con unas fases y rondas determinadas, es por ello que, para poder llevar al mercado la app, se deberán cumplir, al menos, los siguientes objetivos:

- Guardado y edición de datos de usuario
- Módulo de administrador para generar nuevos partidos programados
- Módulo de generación de torneos y equipos

Una vez desarrollados, la aplicación se podrá poner en producción. Hasta entonces, se ofrecerá en modo “test” usar a los componentes de la liga para poder terminar de pulir todos los pequeños detalles que puedan ir surgiendo.

A medio plazo se espera tener una aplicación pequeña, pero estable y funcional.

A largo plazo, conforme se puede apreciar en la memoria, se ha diseñado la aplicación para poder ser la base de algo más viable económicamente, y de la que quizá, con una buena organización futura, se pueda realizar una Start Up.

La inclusión de contenido promocionado, y la búsqueda de nuevas funcionalidades, siempre basadas en el espíritu de “útiles, funcionales y simples” hará de WinnersPadel un proyecto con futuro que podría establecerse en el mercado como una opción real en un nicho descuidado: Las pequeñas competiciones.

5. Glosario

Pádel: Deporte parecido al tenis, que se juega en un reciento cerrado con unas paredes que impiden a la pelota salir.
Android: Sistema operativo desarrollado en la actualidad por Google, que permite a un dispositivo móvil poder ejecutar aplicaciones.
Android Wear: Sistema operativo basado en Android, pero pensado para dispositivos pequeños, como relojes inteligentes.
UML: Es un estándar de diseño de gráficos para poder explicar visualmente funcionalidades, problemas y casuísticas de una aplicación informática.
Diagrama de Gantt: Es un gráfico UML consistente en una planificación detallada de tiempos de trabajo.
DAFO: Es un gráfico que permite, de una forma visual, conocer los diferentes aspectos de un problema dado, pudiendo así valorar como afrontarlo
Login/Logueo: Términos anglosajones usados en el mundo de la informática para expresar el hecho de que un usuario valide sus credenciales.
Torneo: Organización deportiva que permite a una serie de integrantes disputar encuentros de diferentes tipos, para decidir un ganador
Liga: Tipo de torneo que se corresponde con una organización de partidos en la que todos juegan con todos para decidir el ganador
Play-off: Tipo de torneo que se corresponde con una organización de partidos donde dos equipos se enfrentan entre sí, y solo el ganador va superando fases para enfrentarse con otros ganadores, decidiendo en un partido final quien es el vencedor del torneo.
Equipo: Grupo de personas con un objetivo común.
Partido: Enfrentamiento entre 2 personas, o equipos.
Set: Organización de puntos en el pádel. Un partido tiene un máximo de 3 sets siendo estos de un máximo de 6 juegos por equipo, o 7 si se llega a un desempate
Juego: Organización de puntos en el pádel. Un set contiene juegos, que se ganarán al conseguir un determinado número de puntos.
Punto: Organización de puntos en el pádel. Pueden tomar los valores de 0, 15, 30, 40, ventaja.
Tie-break: Desempate en un set cuando ambos equipos han llegado a 6 juegos.
Monetización: Sistema de conversión de funcionalidades de una aplicación a un valor económico que permita una supervivencia en el mercado.
Serverless: Sistema de desarrollo basado en la filosofía de olvidar el tipo de servidor en el que se desplegará, o que tecnologías habrá por debajo. Para ello serverless permite aceptar cualquier tipo de programación quitando al programador la responsabilidad del mantenimiento de una infraestructura en el servidor.
Amazon: Compañía empresarial dedicada al comercio electrónico, que ha desarrollado sistemas informáticos y ofrece el servicio de estos a usuarios.
AWS: Siglas de Amazon Web Service, que se corresponde con los servicios informáticos que ofrece Amazon para sus usuarios.
JWT: Siglas de Json Web Token, que se corresponde con un código alfanumérico único, estructurado en 3 partes lógicas.

NoSQL: Siglas de Not Only SQL, que se corresponde con un concepto de bases de datos no relacionales
JSON: Organización de datos en formato clave-valor para la transmisión de información a través de diferentes dispositivos
Backend: Todo el sistema de tecnologías que componen la parte de procesamiento de datos en una aplicación informática.
Frontend: Todo el sistema de tecnologías que componen la parte de interacción con el usuario en una aplicación informática.
DNS: Siglas de Domain Name System, que corresponde con la tecnología que permite a una petición de internet poder llegar a la máquina que puede atenderla.
Rest: Tipo de petición en internet que sirve para solicitar datos a un servidor, siguiendo un idioma específico
WebService: Servicio de atención de peticiones vía internet en una máquina que permite realizar una funcionalidad en el backend y devolver un resultado.
Caché: Memoria interna de un dispositivo, dedicada a almacenar de forma física, contenido que se borrará ya que no es necesario mantener, pero que permite mejorar la velocidad de consulta mientras la aplicación está activa.
Debugar: Recorrer el código de una aplicación, línea a línea para poder conocer el funcionamiento y solucionar problemas.

6. Bibliografía

Todas las imágenes de este documento han sido procesadas manualmente u obtenidas de forma legal en la web https://pixabay.com

[0] https://pixabay.com/es/illustrations/copa-podio-trofeo-oro-plata-1615074 (visitado 25/10/2019)

[1] https://www.digitalinformationworld.com/2019/05/android-pie-10-percent-users-still-behind-ios.html (visitado 09/10/2019)

[2] https://cincodias.elpais.com/cincodias/2017/08/03/fortunas/1501772874_868688.html (visitado 09/10/2019)

[3] https://www.returngis.net/2019/04/oauth-2-0-openid-connect-y-json-web-tokens-jwt-que-es-que/ (visitado 19/10/2019)

[4] https://tools.ietf.org/html/rfc7519 (visitado 19/10/2019)

[5] https://docs.aws.amazon.com/es_es/apigateway/latest/developerguide/apigateway-use-lambda-authorizer.html (visitado 23/10/2019)

[6] https://pixabay.com/es/ (visitado desde 15/09/2019 hasta el 03/01/2020)

[7] https://aukera.es/blog/bases-de-datos-relacionales-vs-no-relacionales/ (visitado 24/10/2019)

[8] https://pandorafms.com/blog/es/nosql-vs-sql-diferencias-y-cuando-elegir-cada-una (visitado 24/10/2019)

[9] https://openid.net/connect/ (visitado 26/10/2019)

[10] https://docs.aws.amazon.com/es_es/cognito/latest/developerguide/what-is-amazon-cognito.html (visitado 26/10/2019)

[11] https://aws.amazon.com/es/what-is-aws/ (visitado 09/10/2019)

[12] https://devexperto.com/clean-architecture-android/ (visitado 15/11/20109)

[13] https://www.paradigmadigital.com/dev/introduccion-los-componentes-arquitectura-android-go-clean/ (visitado 15/11/2019)

[14] https://jitpack.io/ (visitado 25/11/2019)

[15] https://devexperto.com/mvvm-vs-mvp/ (visitado 20/11/2019)

[16] https://sdos.es/blog/livedata-todo-sobre-el-componente-de-arquitectura-android (visitado 20/11/2019)

[17] https://medium.com/contraslashsas/retrofit-para-android-desde-0-1c8be830a1af (visitado 30/11/2019)

[18] https://github.com/prolificinteractive/material-calendarview (visitado 28/11/2019)

[19] https://anotherdayanotherbug.wordpress.com/2014/10/27/lombok-o-como-eliminar-codigo-boilerplate/ (visitado 26/11/2019)

[20] https://www.baeldung.com/jackson-vs-gson (visitado 11/12/2019)

[21] https://xandroidmarket.com/how-store-data-locally-an-android-app (visitado 21/11/2019)

[22] https://code.tutsplus.com/es/tutorials/get-wear-os-and-android-talking-exchanging-information-via-the-wearable-data-layer--cms-30986 (visitado 05/12/2019)

[23] https://androidayuda.com/2018/03/31/instalar-adb-fastboot-15-segundos-root/ (visitado 11/12/2019)

[24] https://developer.android.com/training/wearables/apps/debugging?hl=es-419 (visitado 11/12/2019)

[25] https://docs.aws.amazon.com/es_es/cognito/latest/developerguide/amazon-cognito-integrating-user-pools-with-identity-pools.html (visitado 10/10/2019)

[26] https://www.youtube.com/watch?v=sUpTwu7iyGs (visitado 18/10/2019)

[28] BootstrapMade (visitado 14/10/2019)

[29] https://docs.aws.amazon.com/es_es/apigateway/latest/developerguide/welcome.html (visitado 15/10/2019)

7. Anexos

7.1. Diagrama de Gantt Planificado

(página 1 anexos)

7.2. Diagrama de Gantt con datos reales de trabajo PEC2

(página 2 anexos)

7.3. Diagrama de Gantt con datos reales de trabajo PEC3

(página 3 anexos)

7.4. Diagrama de Gantt con datos reales de trabajo PEC4

(página 4 anexos)

7.5. Tablas de la base de datos NoSql en DynamoDB

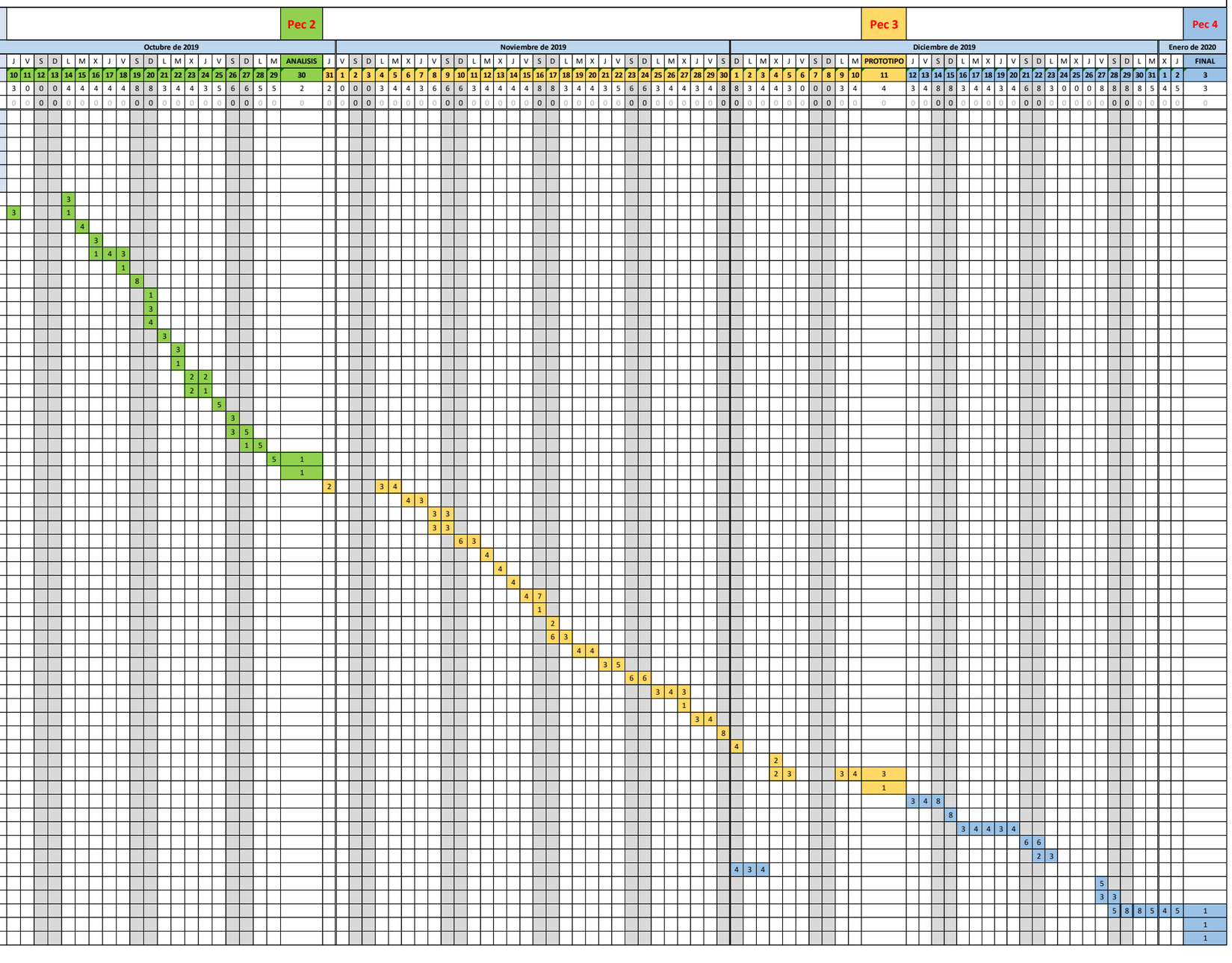
(página 5 anexos)

7.6. Resultado analizado de la encuesta de usuarios

(página 10 anexos)

Planificación TFG

	Horas estimadas	Pec 1	Pec 2	Pec 3	Pec 4
Día Semana / HITO		PROPUESTA	ANALISIS	PROTOTIPO	FINAL
Fechas calendario		9	30	11	3
ESTIMADO de tiempo disponible de trabajo		40	2	4	3
Tiempo restante		0	0	0	0
Lectura del material para iniciar el proyecto	8	8			
Primeros pasos con la memoria.	4	4			
Redacción del primer punto de la memoria	4	4			
Estudio de otras apps parecidas	10	10			
Especificaciones técnicas y funcionales a nivel general	4	4			
Planificación del desarrollo	10	10			
Estudio de usuarios potenciales	3				
Diseño a lápiz de pantallas de app móvil y de reloj	4				
Redacción de funcionalidades de las pantallas del móvil	4				
Redacción de funcionalidades en las pantallas del reloj	3				
Redacción del libro de estilo de la app y sus logos	8				
Redacción de objetivos de seguridad en la aplicación	1				
Redacción de la estructura del servidor	8				
Redacción explicando las versiones escogidas	1				
Identificación de actores	3				
Diagrama de casos de uso	4				
Diagrama de implementación	3				
Diagrama de actividad	3				
Generación del proyecto en Android Studio y sistemas de versionado.	1				
Generación de datos iniciales (usuarios, equipos) para la BD	4				
Alojamiento inicial de datos en la BD DynamoDB de Amazon	3				
Generación de un microservicio de consulta de datos	5				
Securización del microservicio mediante JWT	3				
Alojamiento y testeo del microservicio en Lambda AWS de Amazon	8				
Pantalla de login en la aplicación móvil	6				
Corrección de incidencias detectadas	6				
Firmado de la app y subida a google play de la versión alfa	1				
Carga de datos según rol / trabajo con roles	9				
Microservicio de gestión de equipos	7				
Microservicio de gestión de calendario	6				
Microservicio de gestión de resultados	6				
Pantalla y controlador de inicio en la aplicación móvil	9				
Pantalla y controlador de listado de usuarios/equipos en la app móvil	4				
Pantalla y controlador del calendario/resultados en la aplicación móvil	4				
Pantallas y controladores de administración en la aplicación móvil	4				
Resto de pantallas y controladores derivadas del estudio del diseño	11				
Generación de la librería de conteo de puntos de pádel	1				
Inclusión de la librería de conteo de puntos al móvil	2				
Pantalla y controlador de conteo de puntos en la aplicación móvil	9				
Pantalla y controlador para incluir resultados en la aplicación móvil	8				
Inclusión de la tecnología de notificaciones	8				
Generación de notificaciones y testeo	12				
Pantallas y controladores de navegación de la aplicación de reloj	10				
Inclusión de la librería de conteo de puntos al reloj	1				
Pantalla y controlador de conteo de puntos del reloj	7				
Conexión entre movil y reloj	8				
Conexión entre usuarios de un mismo partido	4				
Testeo de funcionamiento de un partido real	2				
Corrección de errores del testeo	15				
Subida a google play de la versión beta	1				
Generación del paquete de pruebas de la aplicación	15				
Correción de incidencias detectadas	8				
Redacción de la memoria con el desarrollo finalizado	18				
Redacción de la explicación de clases y métodos importantes	12				
Diagrama de clases y secuencia de las partes destacadas	5				
StoryBoard video de presentación promocional y de tribunal	11				
Realización del video explicativo de la aplicación en formato promoción	5				
Realización del video explicativo de la aplicación para el tribunal	6				
Últimas correciones de la aplicación móvil, de reloj , videos y la memoria	36				
Inclusión de la app definitiva en google play	1				
Presentación final de trabajo, aplicaciones movil y de reloj, y videos.	1				
Presentación final de trabajo, aplicaciones movil y de reloj, y videos.	388				



Leyenda
Todos los números son indicaciones en HORAS.
GRIS: primera PEC
VERDE: segunda PEC
AMARILLO: tercera PEC
AZUL: cuarta PEC

Planificación TFM

	Horas estimadas	Pec 1	Pec 2	Pec 3	Pec 4
Día Semana / HITO		PROPUESTA	ANALISIS	PROTOTIPO	FINAL
Fechas calendario		10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 1 2 3
ESTIMADO de tiempo disponible de trabajo	348	3 0 0 0 4 4 4 4 4 8 8 3 4 4 3 5 6 5 5 2 2 0 0 0 3 4 4 3 6 6 6 3 4 4 4 4 8 8 3 4 4 3 5 6 6 3 4 4 3 4 8 8 3 4 4 3 0 0 0 3 4 4 3 4 8 8 3 4 4 3 4 6 8 3 0 0 0 8 8 8 8 5 4 5 3
Tiempo REAL de trabajo	357	5 0 0 1 1 3 3 5 4 7 7 4 6 7 4 5 6 1 3 3 0 6 8 4 0 3 4 4 3 6 6 6 3 4 4 4 4 8 8 3 4 4 3 5 6 6 3 4 4 3 4 8 8 3 4 4 3 0 0 0 3 4 4 3 4 8 8 3 4 4 3 4 6 8 3 0 0 0 8 8 8 8 5 4 5 3
~~Lectura del material para iniciar el proyecto~~	8	8			
~~Primeros pasos con la memoria.~~	4	4			
~~Redacción del primer punto de la memoria~~	4	4			
~~Estudio de otras apps parecidas~~	10	10			
~~Especificaciones técnicas y funcionales a nivel general~~	4	4			
~~Planificación del desarrollo~~	10	10			
~~Estudio de usuarios potenciales~~	8		3 5		
~~Diseño de pantallas de app móvil y de reloj~~	14		2 1 1 3 3 2 1 1		
~~Redacción de funcionalidades de las pantallas del móvil~~	4		1 2 1		
~~Diseño de la monetización~~	3		3		
~~Redacción de funcionalidades en las pantallas del reloj~~	1		1		
~~Búsqueda de imágenes y edición de fotos para la memoria y apliación~~	3		1 1 1		
~~Estudio y redacción de objetivos de seguridad en la aplicación~~	4		1 1 2		
~~Estudio de AWS, y todo su ecosistema (S3, cognito, Lambda...)~~	18		4 6 1 1 1 1 1 1 1 1		
~~Correcciones en la redación de la memoria, y restructuración de la misma~~	2		1 1		
~~Identificación de actores, casos de uso~~	4		1 3		
~~Escenarios de uso~~	2		2		
~~Diagrama de implementación~~	1		1		
~~Diagrama de actividad~~	2		2		
~~Generación del proyecto en Android Studio y sistemas de versionado.~~	1		1		
~~Generación de datos iniciales (usuarios, equipos) para la BD~~	1		1		
~~Estudio de DynamoDB y de la solución para almacenar los datos de usuario~~	2		1 1		
~~Estudio, diseño y redacción de la estructura de la base de datos~~	5		3 2		
Generación del API Gateway de Amazón	2		2		
Conexión con Cognito	2		2		
Alojamiento y testeo del microservicio en Lambda AWS de Amazon	2		2		
Pantalla de login en la aplicación móvil	6		6		
Corrección de incidencias detectadas	2		2		
Firmado de la app y subida a google play de la versión alfa	2		2		
Carga de datos según rol / trabajo con roles	9		2 3 4		
Microservicio de gestión de equipos	7		4 3		
Microservicio de gestión de calendario	6		3 3		
Microservicio de gestión de resultados	6		3 3		
Pantalla y controlador de inicio en la aplicación móvil	9		6 3		
Pantalla y controlador de listado de usuarios/equipos en la app móvil	4		4		
Pantalla y controlador del calendario/resultados en la aplicación móvil	4		4		
Pantallas y controladores de administración en la aplicación móvil	4		4		
Resto de pantallas y controladores derivadas del estudio del diseño	11		4 7		
Generación de la librería de conteo de puntos de pádel	1		1		
Inclusión de la librería de conteo de puntos al móvil	2		2		
Pantalla y controlador de conteo de puntos en la aplicación móvil	9		6 3		
Pantalla y controlador para incluir resultados en la aplicación móvil	8		4 4		
Inclusión de la tecnología de notificaciones	8		3 5		
Generación de notificaciones y testeo	12		6 6		
Pantallas y controladores de navegación de la aplicación de reloj	10		3 4 3		
Inclusión de la librería de conteo de puntos al reloj	1		1		
Pantalla y controlador de conteo de puntos del reloj	7		3 4		
Conexión entre movil y reloj	8		8		
Conexión entre usuarios de un mismo partido	4		4		
Testeo de funcionamiento de un partido real	2		2		
Corrección de errores del testeo	15		2 3 3 4	3	
Subida a google play de la versión beta	1			1	
Generación del paquete de pruebas de la aplicación	15		3 4 8		
Correción de incidencias detectadas	8		8		
Redacción de la memoria con el desarrollo finalizado	18		3 4 4 4 3 4		
Redacción de la explicación de clases y métodos importantes	12		6 6		
Diagrama de clases y secuencia de las partes destacadas	5		2 3		
StoryBoard video de presentación promocional y de tribunal	11		4 3 4		
Realización del video explicativo de la aplicación en formato promoción	5		5		
Realización del video explicativo de la aplicación para el tribunal	6		3 3		
Últimas correciones de la aplicación móvil, de reloj , videos y la memoria	36		5 8 8 5 4 5		1
Inclusión de la app definitiva en google play	1				1
Presentación final de trabajo, aplicaciones movil y de reloj, y videos.	1				1
Presentación final de trabajo, aplicaciones movil y de reloj, y videos.	397				

Leyenda
Todos los números son indicaciones en HORAS.
GRIS: primera PEC
VERDE: segunda PEC
AMARILLO: tercera PEC
AZUL: cuarta PEC

Trabajo Final de Máster - Máster universitario en Desarrollo de aplicaciones para dispositivos móviles - Winners. Aplicación para gestionar una liga de pádel desde el móvil y el reloj

Planificación TFM

	Horas estimadas
~~Lectura del material para iniciar el proyecto~~	8
~~Primeros pasos con la memoria.~~	4
~~Redacción del primer punto de la memoria~~	4
~~Estudio de otras apps parecidas~~	10
~~Especificaciones técnicas y funcionales a nivel general~~	4
~~Planificación del desarrollo~~	10
~~Estudio de usuarios potenciales~~	8
~~Diseño de pantallas de app móvil y de reloj~~	14
~~Redacción de funcionalidades de las pantallas del móvil~~	4
~~Diseño de la monetización~~	3
~~Redacción de funcionalidades en las pantallas del reloj~~	1
~~Búsqueda de imágenes y edición de fotos para la memoria y apliación~~	3
~~Estudio y redacción de objetivos de seguridad en la aplicación~~	4
~~Estudio de AWS, y todo su ecosistema (S3, cognito, Lambda...)~~	18
~~Correcciones en la redación de la memoria, y restructuración de la misma~~	2
~~Identificación de actores, casos de uso~~	4
~~Escenarios de uso~~	2
~~Diagrama de implementación~~	1
~~Diagrama de actividad~~	2
~~Generación del proyecto en Android Studio y sistemas de versionado.~~	1
~~Generación de datos iniciales (usuarios, equipos) para la BD~~	1
~~Estudio de DynamoDB y de la solución para almacenar los datos de usuario~~	2
~~Estudio, diseño y redacción de la estructura de la base de datos~~	5
~~Generación del API Gateway de Amazón~~	19
~~Estudio y conexion con Cognito para el logueo de usuario de forma segura~~	11
~~Pantalla de login en la aplicación móvil~~	7
Pantalla de nuevo usuario en la aplicación móvil	0
~~Estudio de las funciones Lambda AWS de Amazon~~	36
~~Cargar datos de usuarios y aplicación en DynamoDB para hacer pruebas~~	5
~~Conexión con DynamoDB desde Lamba para mostrar datos~~	18
~~Rediseño y programación de pantallas para adaptar mejor las funcionalidades~~	8
~~Pantalla y controlador de inicio en la aplicación móvil~~	4
~~Pantalla y controlador de listado de usuarios/equipos en la app móvil~~	22
~~Pantalla y controlador del calendario/resultados en la aplicación móvil~~	20
Pantallas y controladores de administración en la aplicación móvil	0
~~Resto de pantallas y controladores derivadas del estudio del diseño~~	25
~~Inclusión de la librería de conteo de puntos al móvil~~	6
~~Redacción de la memoria~~	20
~~Pantalla y controlador de conteo de puntos en la aplicación móvil~~	5
~~Pantalla y controlador para incluir resultados en la aplicación móvil~~	6
~~Inclusión de la tecnología de notificaciones~~	0
Generación de notificaciones y testeo	0
~~Controladores de navegación de la aplicación de reloj~~	5
~~Inclusión de la librería de conteo de puntos al reloj~~	1
~~Pantalla y controlador de conteo de puntos del reloj~~	1
~~Conexión entre movil y reloj~~	3
Conexión entre usuarios de un mismo partido	1
~~Testeo de funcionamiento de un partido real~~	2
~~Corrección de errores del testeo~~	8
Subida a google play de la versión beta	0
Generación del paquete de pruebas de la aplicación	15
Correción de incidencias detectadas	8
Redacción de la memoria con el desarrollo finalizado	18
Redacción de la explicación de clases y métodos importantes	12
Diagrama de clases y secuencia de las partes destacadas	5
StoryBoard video de presentación promocional y de tribunal	3
Realización del video explicativo de la aplicación en formato promoción	5
Realización del video explicativo de la aplicación para el tribunal	6
Últimas correciones de la aplicación móvil, de reloj , videos y la memoria	36
Inclusión de la app definitiva en google play	1
Presentación final de trabajo, aplicaciones movil y de reloj, y videos.	1
Presentación final de trabajo, aplicaciones movil y de reloj, y videos.	**458**

Pec 1 · Pec 2 · Pec 3 · Pec 4

Octubre de 2019

	PROPUESTA	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	ANALISIS
Día Semana / HITO																						
Fechas calendario		10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
ESTIMADO de tiempo disponible de trabajo	348	3	0	0	0	4	4	4	4	4	8	8	3	4	4	3	5	6	6	5	5	2
Tiempo REAL de trabajo	418	5	0	0	1	1	3	3	5	4	7	7	4	6	7	4	5	6	1	3	3	0

Noviembre de 2019

	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S
Día Semana / HITO																															
Fechas calendario	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
ESTIMADO de tiempo disponible de trabajo	2	2	0	0	0	3	4	4	3	6	6	6	3	4	4	4	4	8	8	3	4	4	3	5	6	6	3	4	4	3	4
Tiempo REAL de trabajo	3	0	0	0	8	6	5	6	4	5	0	0	5	4	6	6	6	9	11	5	7	4	5	5	9	9	6	7	7	5	8

Diciembre de 2019

	D	L	M	X	J	V	S	D	L	M	PROTOTIPO	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M
Día Semana / HITO																															
Fechas calendario	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
ESTIMADO de tiempo disponible de trabajo	8	3	4	4	3	0	0	0	3	4	4	3	4	3	4	3	4	4	8	3	4	3	4	3	4	6	8	3	0	0	0
Tiempo REAL de trabajo	0	3	6	6	8	8	11	7	6	6	5	4	5	8	8	3	4	4	3	4	6	8	3	0	0	0	8	8	8	8	5

Enero de 2020

	J	FINAL
Día Semana / HITO		
Fechas calendario	1	2	3
ESTIMADO de tiempo disponible de trabajo	4	5	3
Tiempo REAL de trabajo	4	5	3

Leyenda
Todos los números son indicaciones en HORAS.
GRIS: primera PEC
VERDE: segunda PEC
AMARILLO: tercera PEC
AZUL: cuarta PEC

Trabajo Final de Máster - Máster universitario en Desarrollo de aplicaciones para dispositivos móviles - Winners. Aplicación para gestionar una liga de pádel desde el móvil y el reloj

Planificación TFM

	Horas estimadas
Lectura del material para iniciar el proyecto	8
Primeros pasos con la memoria.	4
Redacción del primer punto de la memoria	4
Estudio de otras apps parecidas	10
Especificaciones técnicas y funcionales a nivel general	4
Planificación del desarrollo	10
Estudio de usuarios potenciales	8
Diseño de pantallas de app móvil y de reloj	14
Redacción de funcionalidades de las pantallas del móvil	4
Diseño de la monetización	3
Redacción de funcionalidades en las pantallas del reloj	1
Búsqueda de imágenes y edición de fotos para la memoria y apliación	3
Estudio y redacción de objetivos de seguridad en la aplicación	4
Estudio de AWS, y todo su ecosistema (S3, cognito, Lambda...)	18
Correcciones en la redación de la memoria, y restructuración de la misma	2
Identificación de actores, casos de uso	4
Escenarios de uso	2
Diagrama de implementación	1
Diagrama de actividad	2
Generación del proyecto en Android Studio y sistemas de versionado.	1
Generación de datos iniciales (usuarios, equipos) para la BD	1
Estudio de DynamoDB y de la solución para almacenar los datos de usuario	2
Estudio, diseño y redacción de la estructura de la base de datos	5
Generación del API Gateway de Amazón	19
Estudio y conexion con Cognito para el logueo de usuario de forma segura	11
Pantalla de login en la aplicación móvil	7
Pantalla de nuevo usuario en la aplicación móvil	0
Estudio de las funciones Lambda AWS de Amazon	36
Cargar datos de usuarios y aplicación en DynamoDB para hacer pruebas	5
Conexión con DynamoDB desde Lamba para mostrar datos	18
Rediseño y programación de pantallas para adaptar mejor las funcionalidades	8
Pantalla y controlador de inicio en la aplicación móvil	4
Pantalla y controlador de listado de usuarios/equipos en la app móvil	22
Pantalla y controlador del calendario/resultados en la aplicación móvil	20
Pantallas y controladores de administración en la aplicación móvil	0
Resto de pantallas y controladores derivadas del estudio del diseño	25
Inclusión de la librería de conteo de puntos al móvil	6
Redacción de la memoria	20
Pantalla y controlador de conteo de puntos en la aplicación móvil	5
Pantalla y controlador para incluir resultados en la aplicación móvil	6
Inclusión de la tecnología de notificaciones	0
Generación de notificaciones y testeo	0
Controladores de navegación de la aplicación de reloj	5
Inclusión de la librería de conteo de puntos al reloj	1
Pantalla y controlador de conteo de puntos del reloj	1
Conexión entre movil y reloj	3
Conexión entre usuarios de un mismo partido	1
Testeo de funcionamiento de un partido real	2
Corrección de errores del testeo	8
Subida a google play de la versión beta	0
Correción de incidencias detectadas	33
StoryBoard video de presentación promocional y de tribunal	2
Diseño, grabación y maquetación del video de Elevator Pitch	10
Diseño, grabación y maquetación del video de como funciona la app	16
Diseño, grabación y maquetación del video de cómo es AWS	6
Diseño, grabación y maquetación del video de ejemplo en un partido	20
Últimas correciones de la aplicación móvil, de reloj , videos y la memoria	9
Inclusión de la app definitiva en google play	4
Redacción de la memoria con el desarrollo finalizado	5
Presentación final de trabajo, aplicaciones movil y de reloj, y videos.	1
Presentación final de trabajo, aplicaciones movil y de reloj, y videos.	**454**

Leyenda
Todos los números son indicaciones en HORAS.
GRIS: primera PEC
VERDE: segunda PEC
AMARILLO: tercera PEC
AZUL: cuarta PEC

Tablas de la base de datos NoSql en DynamoDB

cognito	Servicio de AWS para almacenar, consultar, autenticar y autorizar los datos de usuario
Atributo	**Tipo dato**	**Descripción**
sub	String	id en winners
email	String	email
email_verified	Boolean	email verificado (boolean)
nickname	String	nick escogido (no cambia)
given_name	String	nombre de pila
preferred_username	String	nombre que mostrara el jugador
family_name	String	apellidos
phone_number	String	numero de teléfono (con formato internacional -> +34)
phone_number_verified	Boolean	numero de teléfono verificado
birthdate	Date	fecha de nacimiento en formato AAAA-MM-DD
gender	String	genero
locale	String	código de idioma ES_es , ES_ca , EN_en ...
picture	String	imagen de perfil
updated_at	Timestamp	fecha de actualización de estos datos
sys_admin	String	(atributo propio) identifica al usuario como administrador de winners
grpd_varified	Boolean	(atributo propio) el usuario a leído y aceptado el RGPD
show_phone_number	Boolean	(atributo propio) indicará si el usuario quiere o no poner de forma pública su teléfono

competition	Competición como objeto base raiz de un torneo
Atributo	**Tipo dato**	**Descripción**
id	Long	identificador de la competición
name	String	nombre de la competición
description	String	descripción o eslogan
date_init	Date	fecha de inicio del torneo
date_end	Date	fecha de fin del torneo
city_competition	String	ciudad en la que se disputa, si solo es una
region_competition	String	región en la que se disputa, (comunidad autónoma, provincia)
country_competition	String	país en el que se disputa, si solo es uno
published	Boolean	(boolean) indica si el torneo es accesible por todos o solo lo puede ver el administrador
privated	Boolean	(boolean) indica si solo se podrá acceder a él con invitación, o será público
registration_closed	Boolean	(boolean) indica si aun se pueden inscribir jugadores o no
finish_competition	Boolean	(boolean) indica si el torneo ya ha terminado o no
id_before_competition	Long	identificador de la competición de la temporada anterior
id_next_competition	Long	identificador de la competición de la siguiente temporada
id_user_created	Long	identificador del usuario que crea la competición
updated_at	Timestamp	fecha de actualización de los datos
winners_team_competition_list	Array	array con los diferentes equipos campeones que puede haber en una competición (campeón total, campeón de rango, premio de liga...)
winners_user_competition_list	Array	array con los diferentes usuarios campeones que puede haber en una competición (campeón total, campeón de rango, premio de liga...)
phase_list	Array	array con las fases que tiene

phase	Fase de la competición. Una fase de grupos tiene 1 Group_phase, una de eliminatorias tendrá tantas como rondas
Atributo	**Tipo dato**	**Descripción**
id	Long	identificador de la fase
name	String	nombre de la fase
description	String	descripción de la fase
date_init	Date	fecha de inicio de la fase
date_end	Date	fecha de fin de la fase
type_playoff	Boolean	(Boolean) indica si esta fase tiene el formato de una eliminatoria, o no y tiene el formato de una liga
group_phase_list	Array	array con las fases grupos que tiene (una competición tipo liga, solo tendrá 1 Group_phase, una competición tipo play-off tendrá tantas Group_phase como rondas, y en ronda los Group tendrán 1 único Match)

group_phase	Fase de grupos. En una competición tipo eliminatorias, este objeto corresponderá a las rondas (8º, 4º…)	
Atributo	**Tipo dato**	**Descripción**
id	Long	identificador de la fase
name	String	nombre de la fase
description	String	descripción de la fase
date_init	Date	fecha de inicio de la fase
date_end	Date	fecha de fin de la fase
group_list	Array	array con los grupos que tiene

group	Objeto que contrendrá un número determinado de equipos para enfrentarse	
Atributo		**Descripción**
id	Long	identificador del grupo
name	String	nombre del grupo
position	Long	posición del grupo en un listado
teams_id_list	Array	array con el id de equipo que tiene (como un equipo es una entidad propia que puede vivir fuera de un grupo específico, aquí solo se almacena la referencia al equipo)
league_day_list	Array	array con las jornadas de un grupo

league_day	Jornada de liga que almacenara varios partidos	
Atributo		**Descripción**
id	Long	identificador de la jornada
number	Long	numero de jornada dentro del grupo
date_init	Date	fecha de inicio de la jornada
date_end	Date	fecha de fin de la jornada
match_list	Array	array con los partidos de una jornada
classification	Object	tendrá la clasificación del grupo

match	Partido entre 2 equipos	
Atributo	**Tipo dato**	**Descripción**
id	Long	identificador del partido
number	Long	número del partido dentro de la jornada
date_match_list	Date	array con las fecha y horas definitivas a la que se jugará el partido (una fecha puede haberse cancelado, y haber buscado otra, el último será el "elegido")
date_proposed_list	Date	array con las propuestas de fecha y hora para jugar el partido
score	Object	(Objeto) marcador del partido
local_id_team	Long	id del equipo que juega como local (se guarda la referencia para poder saber quién lo jugo)
local_name_team	String	nombre que tenía el equipo cuando jugó el partido
local_id_player_1	Long	id del jugador 1 que jugó el partido con el equipo local
local_name_player_1	String	nombre que tenía el jugador 1 cuando jugó el partido con el equipo local
local_id_player_2	Long	id del jugador 2 que jugó el partido con el equipo local
local_name_player_2	String	nombre que tenía el jugador 2 cuando jugó el partido con el equipo local
guest_id_team	Long	id del equipo que juega como visitante (se guarda la referencia para poder saber quién lo jugo)
guest_name_team	String	nombre que tenía el equipo cuando jugó el partido
guest_id_player_1	Long	id del jugador 1 que jugó el partido con el equipo visitante
guest_name_player_1	String	nombre que tenía el jugador 1 cuando jugó el partido con el equipo visitante
guest_id_player_2	Long	id del jugador 2 que jugó el partido con el equipo visitante
guest_name_player_2	String	nombre que tenía el jugador 2 cuando jugó el partido con el equipo visitante
match_location	String	ubicación donde se ha jugado el partido
observations	String	observaciones que se quieran realizar sobre el partido

date_match	Fecha del partido	
Atributo	**Tipo dato**	**Descripción**
id	Long	identificador del partido
date_created	Date	fecha y hora de partido
date_match_definitive	Boolean	indica si a la fecha y hora elegidas hay pista reservada, o no es una hora definitiva
local_confirmed_player_1	Boolean	confirmación por parte del jugador 1 del equipo local, que conoce el partido y asistirá
local_confirmed_player_2	Boolean	confirmación por parte del jugador 2 del equipo local, que conoce el partido y asistirá
guest_confirmed_player_1	Boolean	confirmación por parte del jugador 1 del equipo local, que conoce el partido y asistirá
guest_confirmed_player_2	Boolean	confirmación por parte del jugador 2 del equipo local, que conoce el partido y asistirá
observation	String	explicación, si fuese necesaria de porque la fecha propuesta, porqué ha sido denegada, porqué es una fecha conflictiva o cualquier otra observación

date_proposed	Fecha de propuesta para el partido	
Atributo		**Descripción**
id	Long	identificador de la fecha propuesta
date_proposed	Date	Fecha y hora propuestas
user_id_accepted	Long	array con los id de los usuarios que han aceptado la propuesta
user_id_declined	Long	array con los id de los usuarios que han rechazado la propuesta
proposed_selected	Date	fecha de propuesta seleccionada
proposed_denied	Date	fecha de puesta denegada por algún motivo
observation	String	explicación, si fuese necesaria de porque la fecha propuesta, porqué ha sido denegada, porqué es una fecha conflictiva o cualquier otra observación

score	Marcador del partido	
Atributo	**Tipo dato**	**Descripción**
id	Long	identificador del marcador
id_referee	Long	id del usuario que arbitrará el partido (introducirá los puntos) (puede ser un jugador, o podría ser otro usuario ajeno al partido)
local_id_team	Long	id del equipo que juega como local (se guarda la referencia para poder saber quién lo jugo)
local_name_team	String	nombre que tenía el equipo cuando jugó el partido
local_id_player_1	Long	id del jugador 1 que jugó el partido con el equipo local
local_name_player_1	String	nombre que tenía el jugador 1 cuando jugó el partido con el equipo local
local_id_player_2	Long	id del jugador 2 que jugó el partido con el equipo local
local_name_player_2	String	nombre que tenía el jugador 2 cuando jugó el partido con el equipo local
guest_id_team	Long	id del equipo que juega como visitante (se guarda la referencia para poder saber quién lo jugo)
guest_name_team	String	nombre que tenía el equipo cuando jugó el partido
guest_id_player_1	Long	id del jugador 1 que jugó el partido con el equipo visitante
guest_name_player_1	String	nombre que tenía el jugador 1 cuando jugó el partido con el equipo visitante
guest_id_player_2	Long	id del jugador 2 que jugó el partido con el equipo visitante
guest_name_player_2	String	nombre que tenía el jugador 2 cuando jugó el partido con el equipo visitante
set_list	Object	array de sets del partido
tiempoJuego	Object	Objeto que contiene el tiempo que ha costado jugar el partido
id_team_match_winner	Long	Id del equipo que ha ganado el partido

prize_competition	Premio de la competicion	
Atributo	**Tipo dato**	**Descripción**
id	Long	identificador del tipo de premio
id_team	Long	equipo al que hace referencia
winner_prize	Boolean	(boolean) indica si es el premio oficial del torneo (al campeón final)
name_prize	String	nombre del premio
description_prize	String	descripción del tipo de premio que se lleva

team	Equipo	
Atributo	**Tipo dato**	**Descripción**
id	Long	identificador de la competición
name	String	nombre de la competición
description	String	descripción o eslogan
soft_team	Boolean	Indica si un equipo es "débil" o no. En caso de que lo sea, el equipo solo tendrá vida de 1 jornada, posibilitando así hacer competiciones de carácter individual donde cada partido juegan 2 equipos creados para ese fin, y cuyos puntos se reparten a los jugadores, no al equipo, ya que este desaparecerá cuando termine el partidos
date_created	Date	fecha de creación del equipo
date_deleted	Date	fecha de extinción del equipo
picture	String	imagen de cabecera del equipo
logo	String	imagen de logo del equipo
id_captain_list	Array	array con el id del usuario capitán (en la primera versión solo habrá uno y será el que crea el equipo, en posteriores el capitán podrá nombrar otros)
id_player_list	Array	array con el id de usuario de los jugadores que lo componen (como un usuario es una entidad propia que puede vivir fuera del equipo, aquí solo se almacena la referencia al jugador)
competition_id_list	Array	array de las competiciones que ha disputado el equipo
historic_player_list	Array	array con el los objetos de usuario de los jugadores que dejaron el equipo (solo se guardara al abandonar el equipo) así se mantendrá un histórico

live_score	Marcador en vivo	
Atributo	**Tipo dato**	**Descripción**
id	Long	identificador del marcador en vivo
id_competition	Long	id de la competición en la que está el marcador en vivo
id_match	Long	id del partido al que hace referencia
score	Object	Marcador del partido

set	Set de un partido	
Atributo	**Tipo dato**	**Descripción**
juegosSet	Array	Lista de sets jugados
tiempoJuegoSet	Object	Objeto que contiene el tiempo que ha costado jugar el set
numJuegosVisitante	Long	Numero de juegos que ha conseguido el equipo visitante
juegosVisitante	Array	Listado con los juegos del equipo visitante
numJuegosLocal	Long	Numero de juegos que ha conseguido el equipo local
ganadorJuego	Array	Equipo que ha ganado el juego 1=local, 2=visitante
juegosLocal	Array	Listado con los juegos del equipo local
terminado	Boolean	Indica si el set está terminado

tiempoJuego	Almacenará cuanto tiempo ha costado realizar alguno de los objetos de un partido, set, juego y punto	
Atributo	**Tipo dato**	**Descripción**
horaInicio	Timestamp	Hora de inicio del set
horaFin	Timestamp	Hora de fin del set
tiempoJugadoString	String	Cantidad de tiempo jugado en String
horaFinString	String	Hora fin en String
horaInicioString	String	Hora inicio en String

juegosSet		
Atributo	**Tipo dato**	**Descripción**
puntosJuego	Array	Array de juegos
tiempoJuego	Object	Información temporal del juego
numJuegoDelSet	Long	número de juego dentro del set
ganadorJuego	Long	id del equipo que ha ganado el juego
terminado	Boolean	indica si el set está o no terminado

puntosJuego	Punto de un partido	
Atributo	**Tipo dato**	**Descripción**
valorPuntoLocal	String	Valor del punto (0,15,30,40,AD,=,1,2,3,4,5,6,7)
valorPuntoVisitante	String	Valor del punto (0,15,30,40,AD,=,1,2,3,4,5,6,7)
tiempoDeJuego	Object	Información temporal del juego
ganadorPunto	Long	id del equipo que ha ganado el punto

classification	Tabla de clasificación de un grupo	
Atributo	**Tipo dato**	**Descripción**
id	Long	identificador
classification_team_list	Array	Array de objetos de clasificación

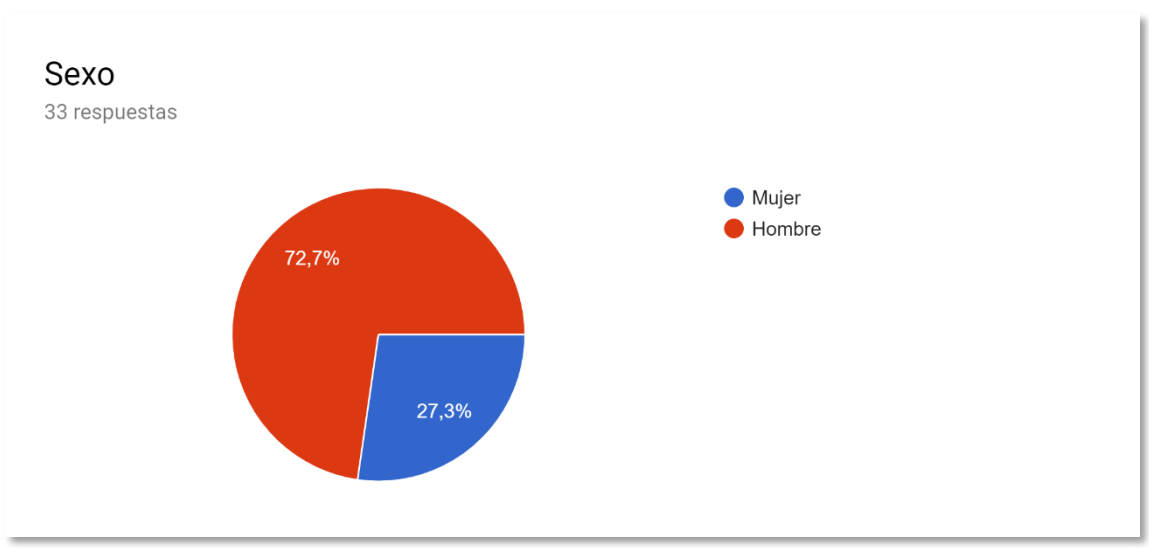
classification_team	Objeto que contendrá loa datos de la tabla de clasificación	
Atributo	**Tipo dato**	**Descripción**
id	Long	identificador del equipo de la clasificación
id_team	Long	identificador del equipo
team_name	String	nombre del equipo
position	Long	posición en la tabla
match_winned	Long	número de partidos ganados
match_equals	Long	número de partidos empatados
match_lost	Long	número de partidos perdidos
sets_win	Long	número de sets ganados
sets_lost	Long	número de sets perdidos
games_win	Long	número de juegos ganados
games_lost	Long	número de juegos perdidos
points	Long	número de puntos conseguidos

player_score	documento que almacenará todos los datos que registre un usuario	
Atributo	**Tipo dato**	**Descripción**
id	Long	identificador del score del usuario
match_winned	Long	número de partidos ganados
match_equals	Long	número de partidos empatados
match_lost	Long	número de partidos perdidos
sets_win	Long	número de sets ganados
sets_lost	Long	número de sets perdidos
games_win	Long	número de juegos ganados
games_lost	Long	número de juegos perdidos

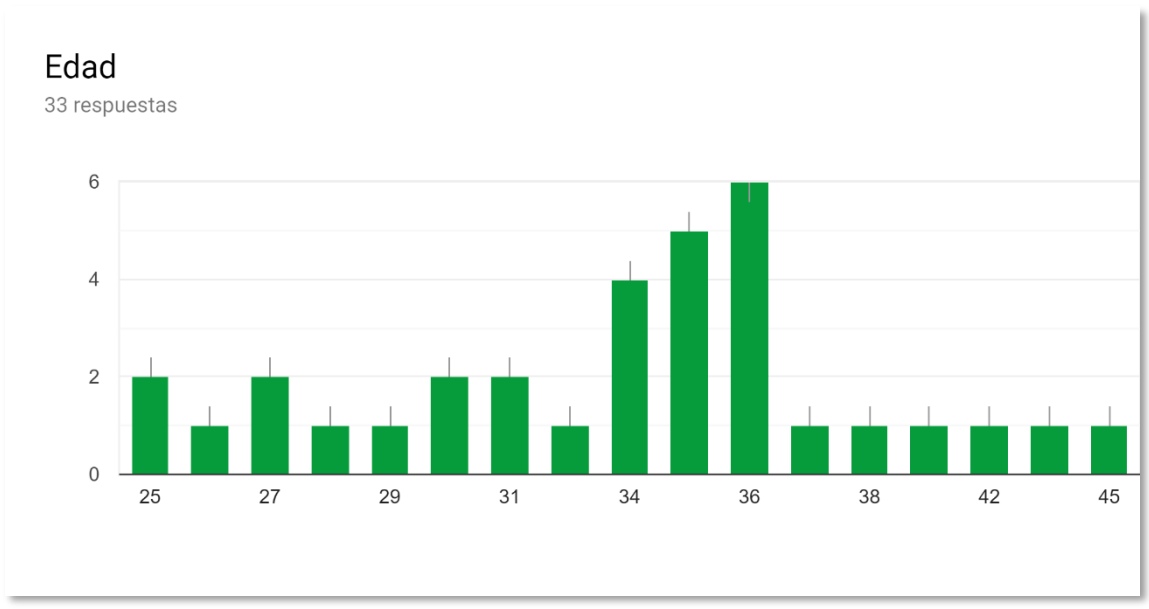
Resultado analizado de la encuesta de usuarios

7.6.1. Información del encuestado

En relación al sexo del encuestado, encontramos una mayoría de hombres

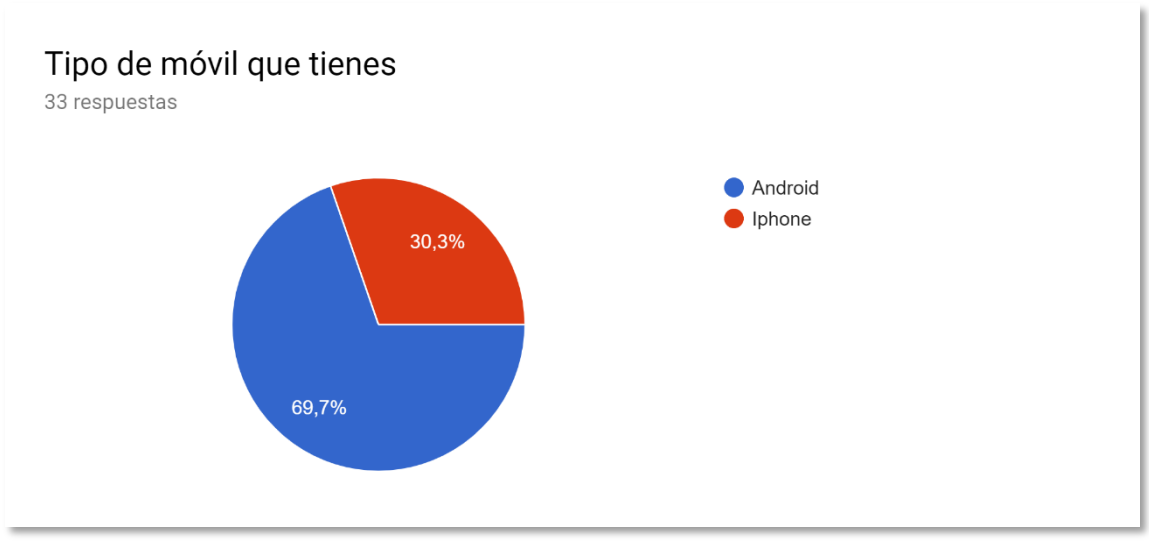


A nivel de información, valoraremos que la mayoría de respuestas vendrán del mundo masculino, dato que, aunque no es fundamental para basar una opinión, sí que conviene tener en cuenta por las diferencias entre sexos existentes en el deporte.

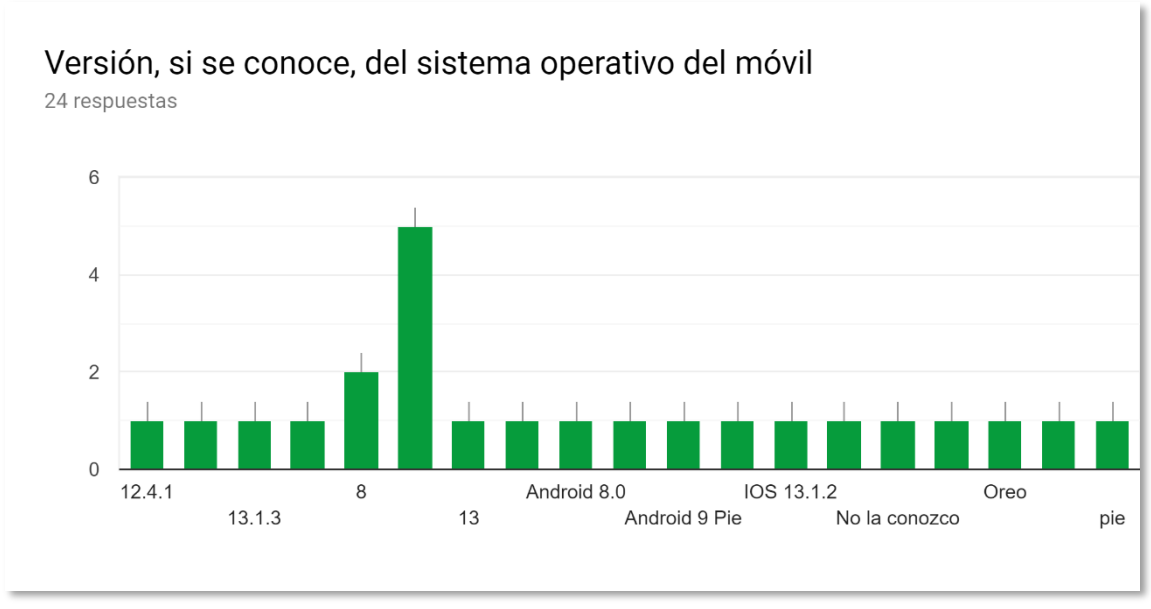


Nos encontramos con que la mayoría de encuestados rondan la treintena de años, por lo que habrá que valorar también los objetivos, capacidades y preocupaciones en la vida de este grupo de edad.

7.6.2. Información sobre el actual dispositivo que tienen



En este caso, podemos ver que la mayoría de ellos tienen un dispositivo Android. Lo cual nos da una idea de con qué dispositivo sacar una primera versión.



Organizados los datos respecto a las versiones que nos dan, encontramos esta información:

Android	nº
android 8	4
android 9	10
android 5	1
android 7	1

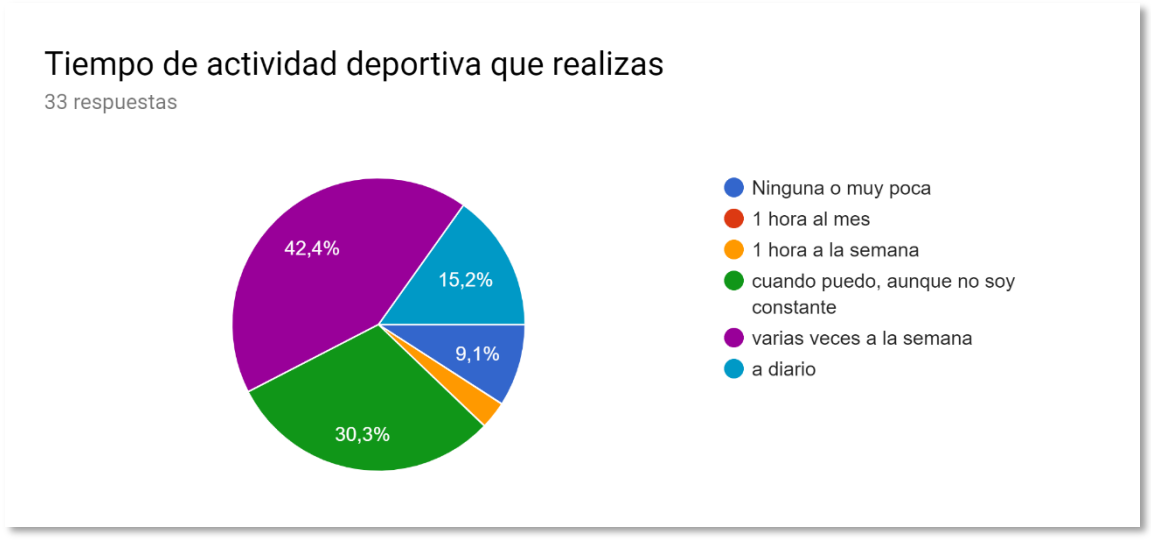
IOS	nº
IOS 10	1
IOS 13	3
IOS 12	2

No la saben
2

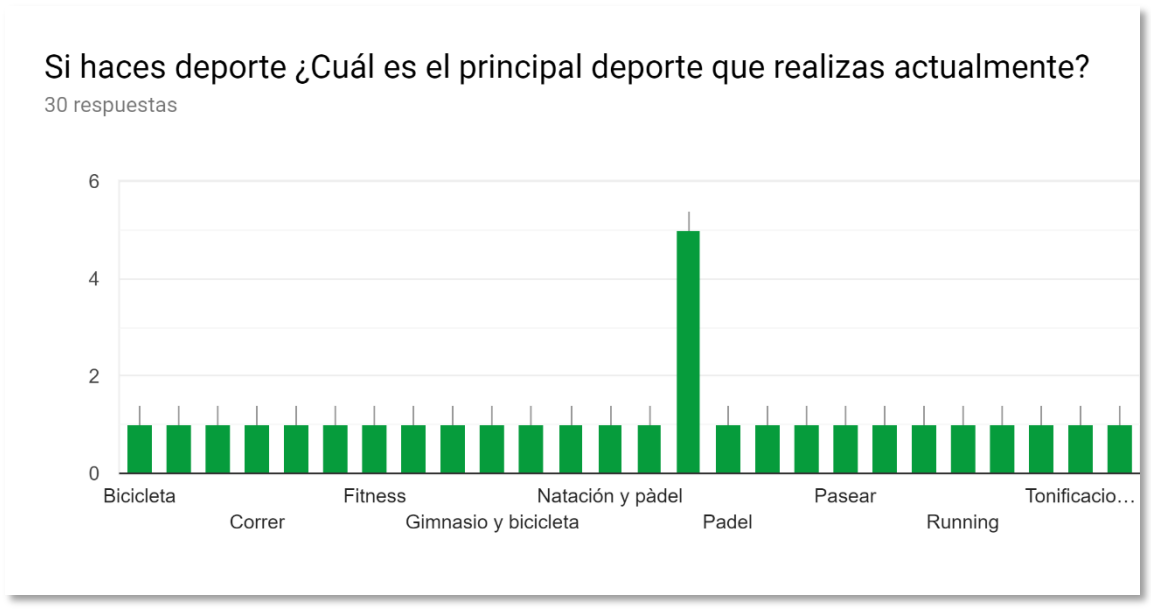
No han constestado
9

Lo que nos da a entender, que tanto en Android como en IOS, la mayoría tienen una versión bastante actual.

7.6.3. Información sobre la vida deportiva y la relación con el pádel

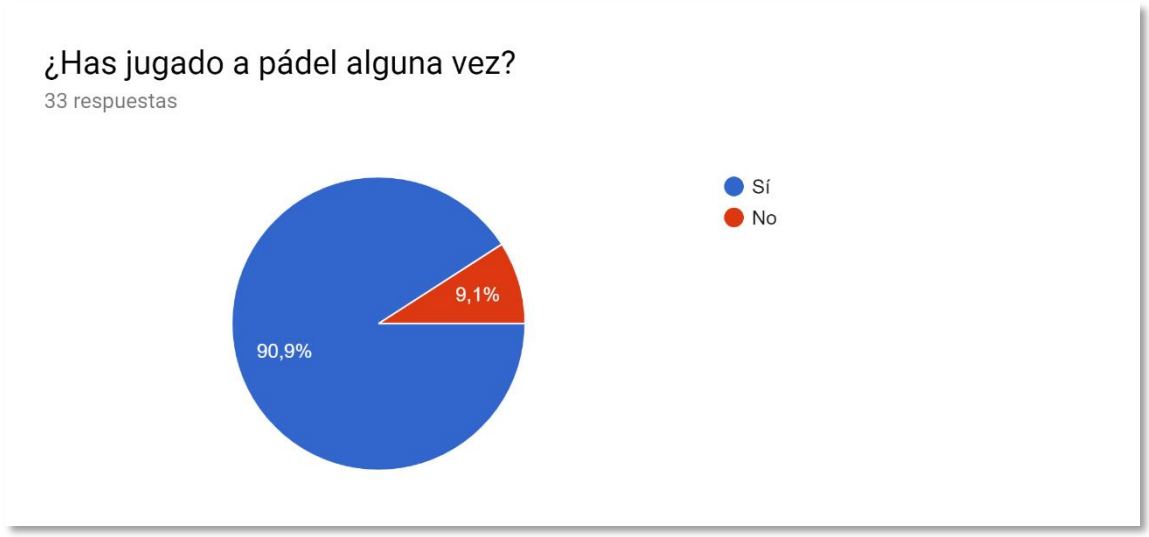


Si juntamos los quesitos más poblados, que son "varias veces a la semana", "a diario", "cuando puedo, aunque no soy constante", nos encontramos con que el 87.9% de los encuestados tienen una vida deportiva bastante activa.

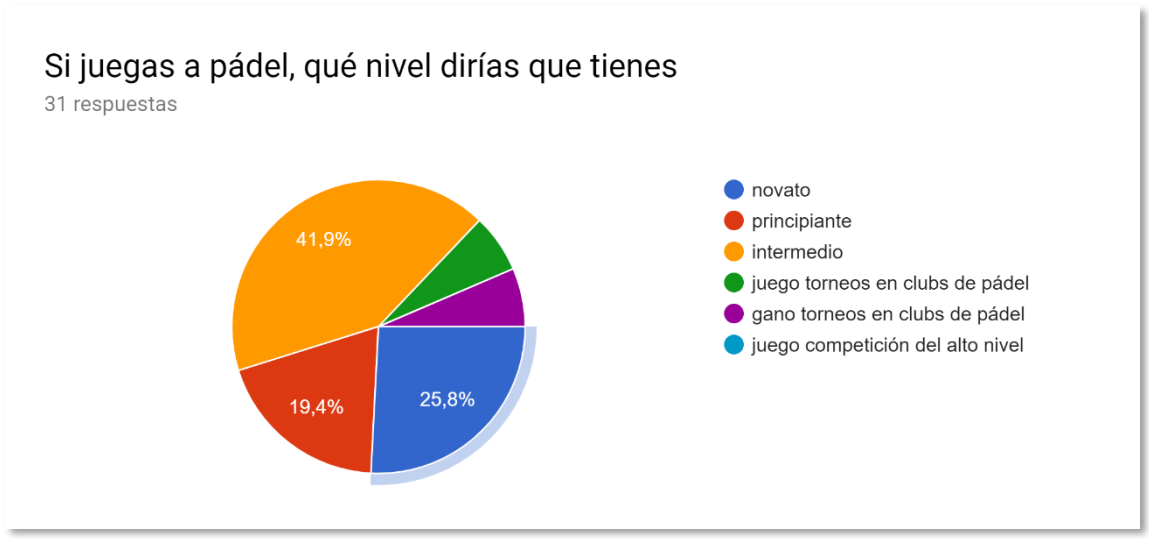


Organizando los datos nos encontramos con la siguiente tabla, donde se puede apreciar que el pádel es el deporte mayoritario entre los encuestados.

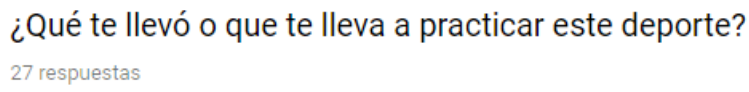
Deporte	nº
Pádel	12
Bicicleta	6
Correr	4
Gym	4
No hago deporte	4
Ioga + Pilates	3
Natación	2
Futbol	2
Andar	1
Trekking	1
Crossfit	1
Tenis	1



Aquí podemos valorar que la gran mayoría ha practicado pádel alguna vez



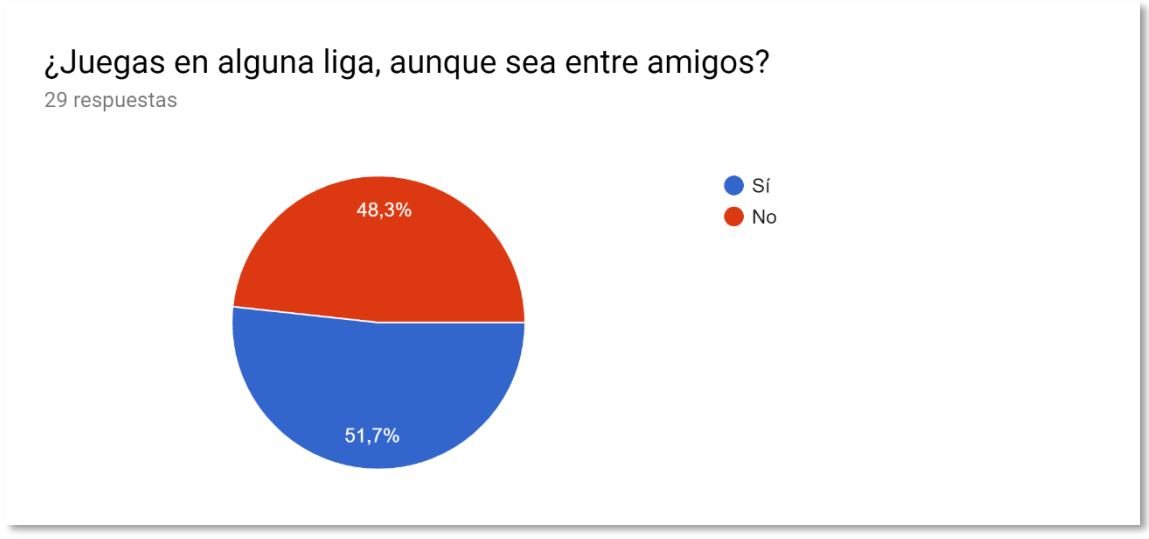
Y de lo que juegan a pádel entre los encuestados, el 45,2% se define con un nivel bajo, un 41,9% con un nivel medio y un 13% con un nivel alto



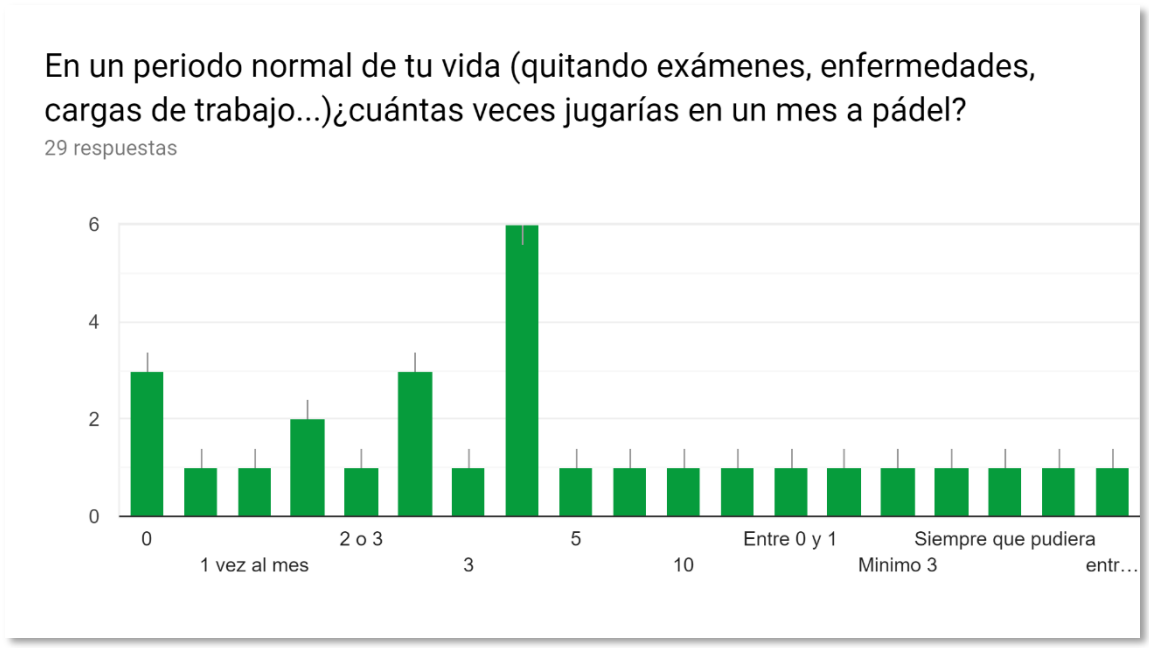
Intentando organizar las respuestas tipo redacción dadas, las podríamos agrupar en los siguientes conceptos;

Tipo de respuesta	nº
Introducido por otras personas	10
Por ocio y diversión	9
Por compromiso con terceros	3
Por curiosidad	2
Fácil y cómodo de realizar	1
Por moda	1

De donde podemos ver que la mayoría ha sido introducida al pádel por terceras personas, o por una motivación de diversión de este deporte.



Está bastante igualado, pero teniendo en cuenta el sesgo de encuestados, podríamos decir que, que un 51,7% estén en una liga es un porcentaje alto.

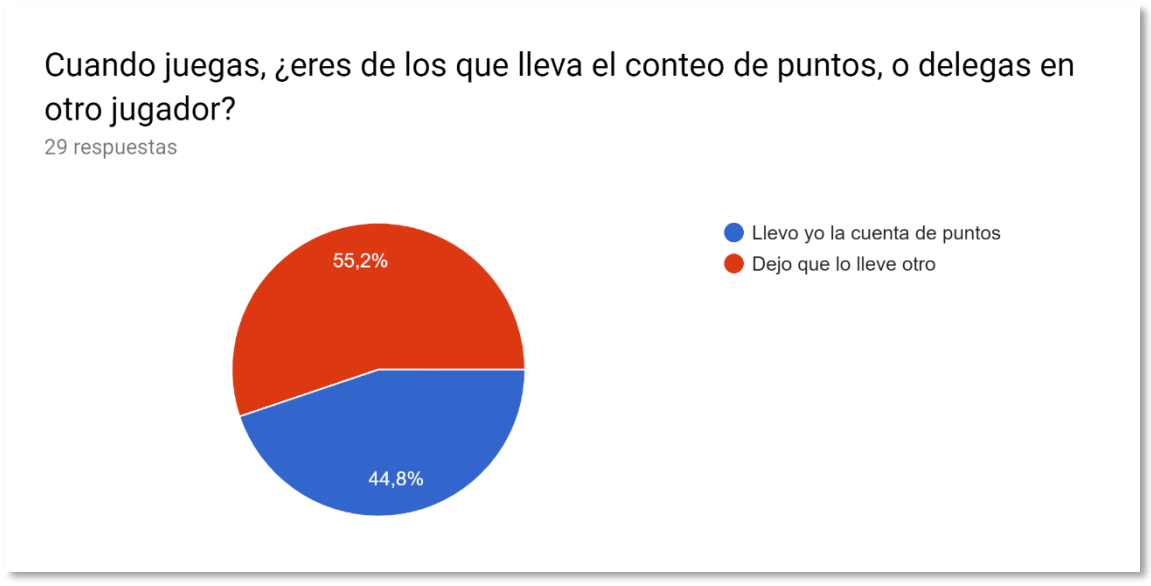


Organizando los datos, agrupando respuestas parecidas, y ordenando luego por grupos, tenemos este resultado

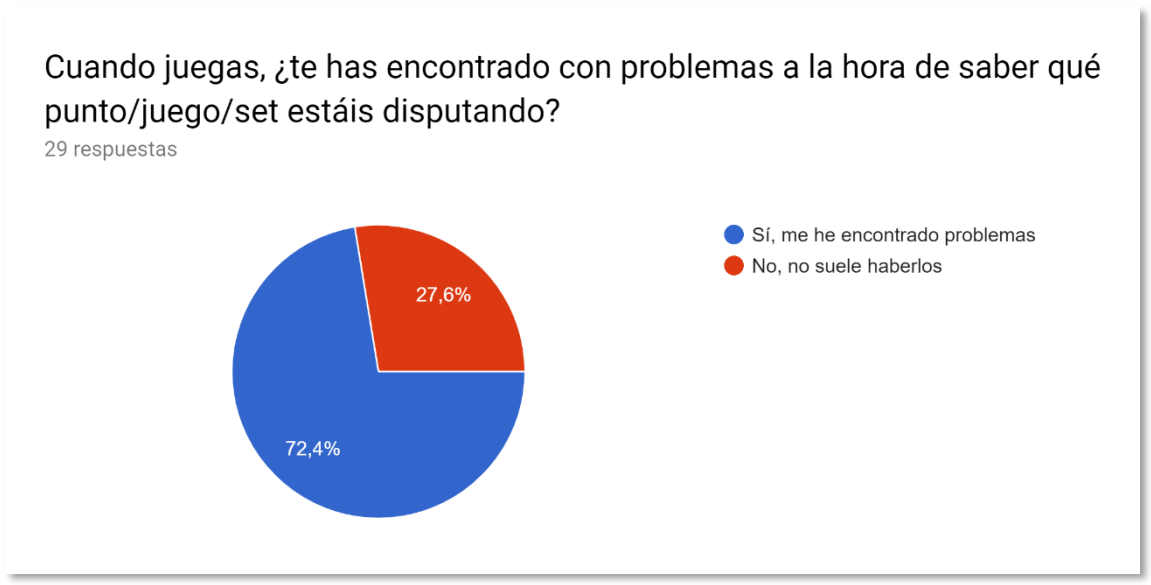
Número de veces	nº
0	4
1	2
3	1
4	6
5	1
6	1
10	1
12	2
0 o 1	2
1 o 2	1
2 o 3	5
4 o más	1
6 o 7	1
Siempre que pudiera	1

Número de veces	nº
0	4
De 1 a 3 veces	11
De 4 a 10 veces	11
Más de 10 veces	3

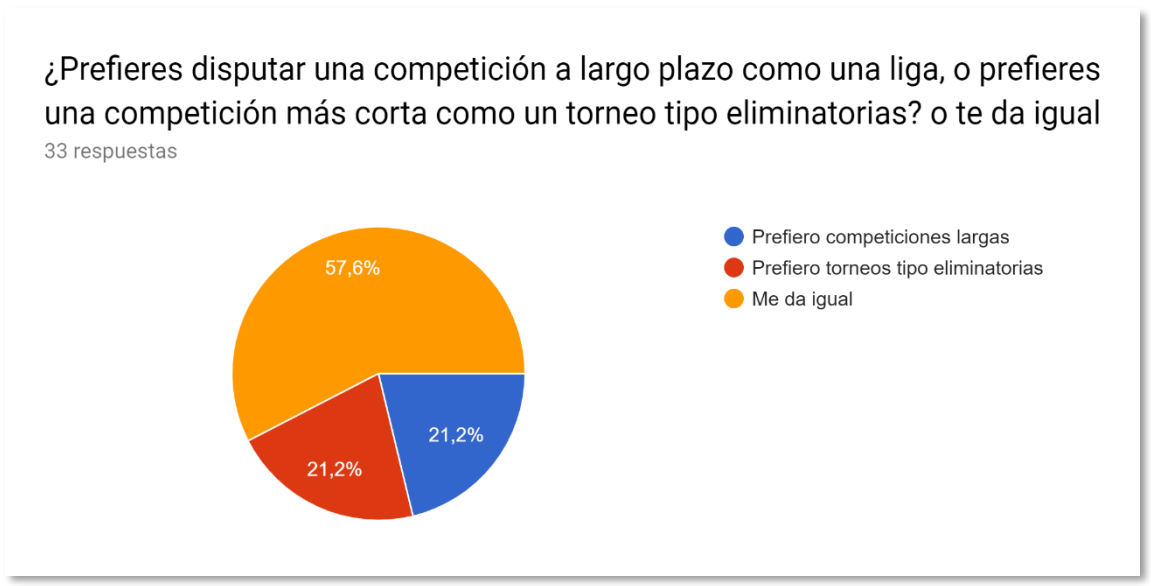
Lo que nos da una idea de que la mayoría de encuestados juega 1 o 2 veces como mucho a la semana, lo que le hace que no sea un fanático de este deporte, pero sí un deportista más o menos asiduo.



La mayoría de encuestados prefiere no llevar la cuenta de los puntos, aunque no es una mayoría aplastante. Podríamos decir que más o menos al 50% hay igualdad entre contar los puntos y no.

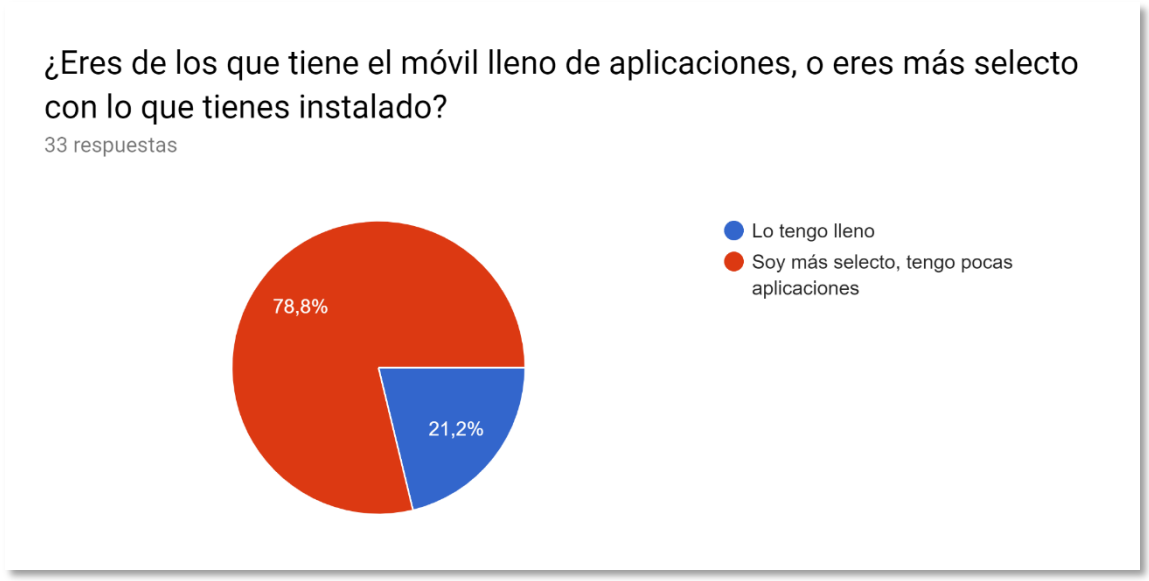


Lleven o no la cuenta de los puntos, casi tres cuartos de los encuestados reconocen que se han encontrado problemas a la hora de contarlos en vivo.



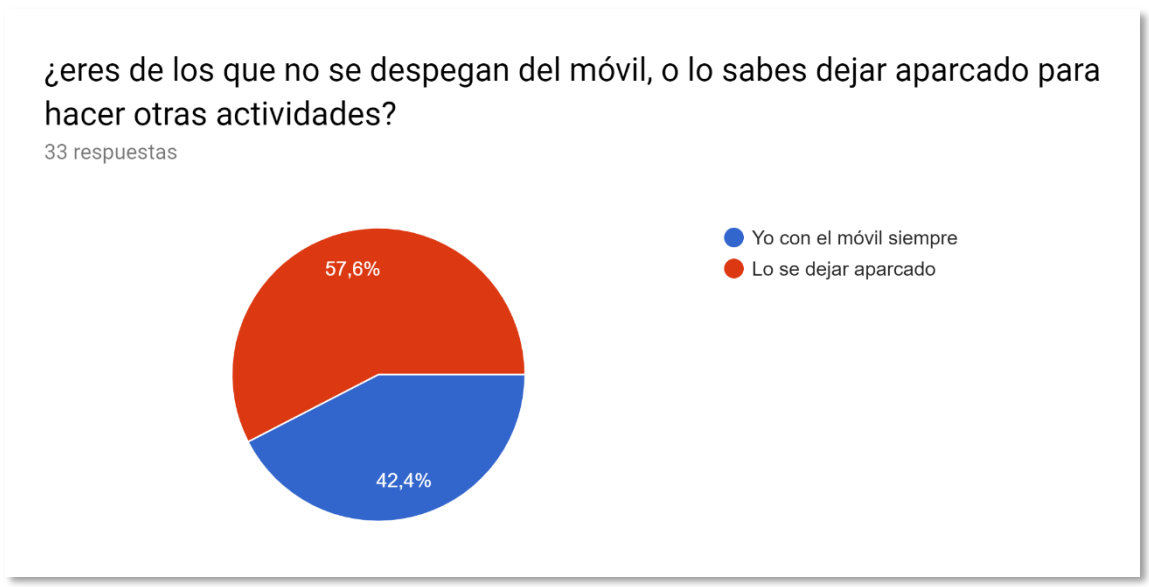
La mayoría de encuestados no tiene preferencia sobre los tipos de competición a disputar, y queda totalmente igualado los que prefieren competiciones tipo liga o copa.

7.6.4. Información sobre la relación que tienen con la tecnología



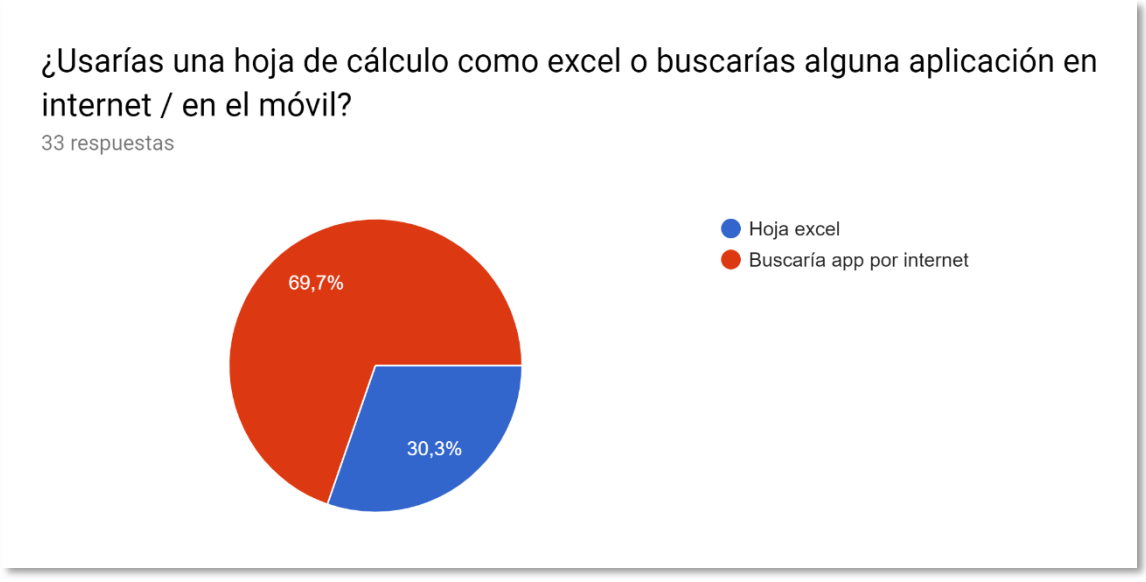
Los encuestados suelen ser bastante selectos a la hora de instalar aplicaciones, por lo que no instalan cualquier cosa, o solo instalan aplicaciones útiles

Verificando el punto anterior, antes que instalar una aplicación, los encuestados prefieres consultarla vía web.

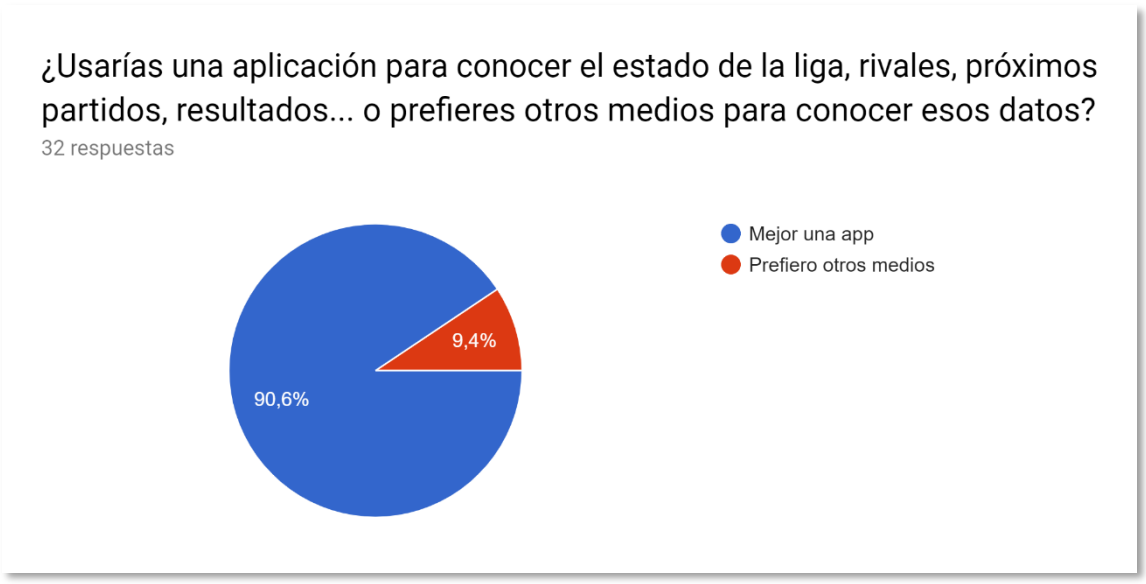


Y en la mayoría de casos, los encuestados no se declaran dependientes del móvil.

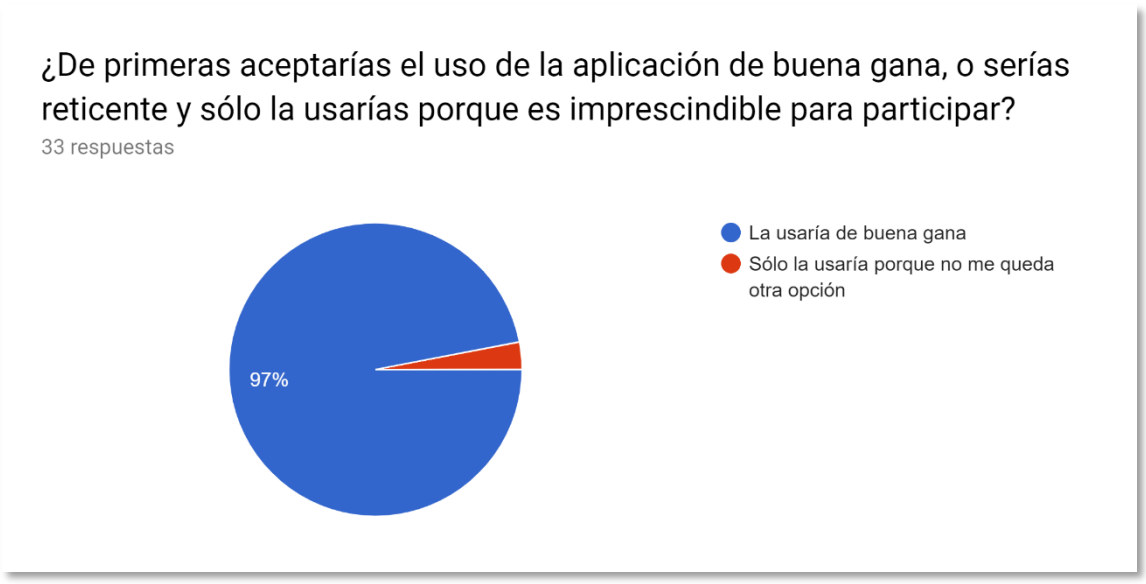
7.6.5. Información relacionada con la futura aplicación y su uso



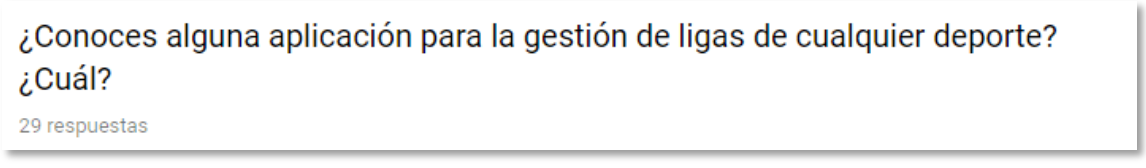
Reconocen los encuestados que, en caso de tener que controlar una liga de pádel, preferirían usar una aplicación, antes que una hoja Excel, aunque tampoco es un recurso descartado.



Y relacionado con la respuesta anterior, también reconocen que, si no la gestionasen, pero tuviesen que participar de una, la gran mayoría preferiría tener una aplicación para consultar los datos.

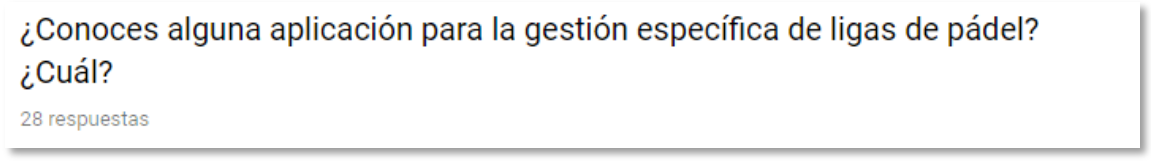


Todas las personas que han respondido al formulario, excepto una, usarían de buena gana esta app y no se sentirían impuestos a su uso.



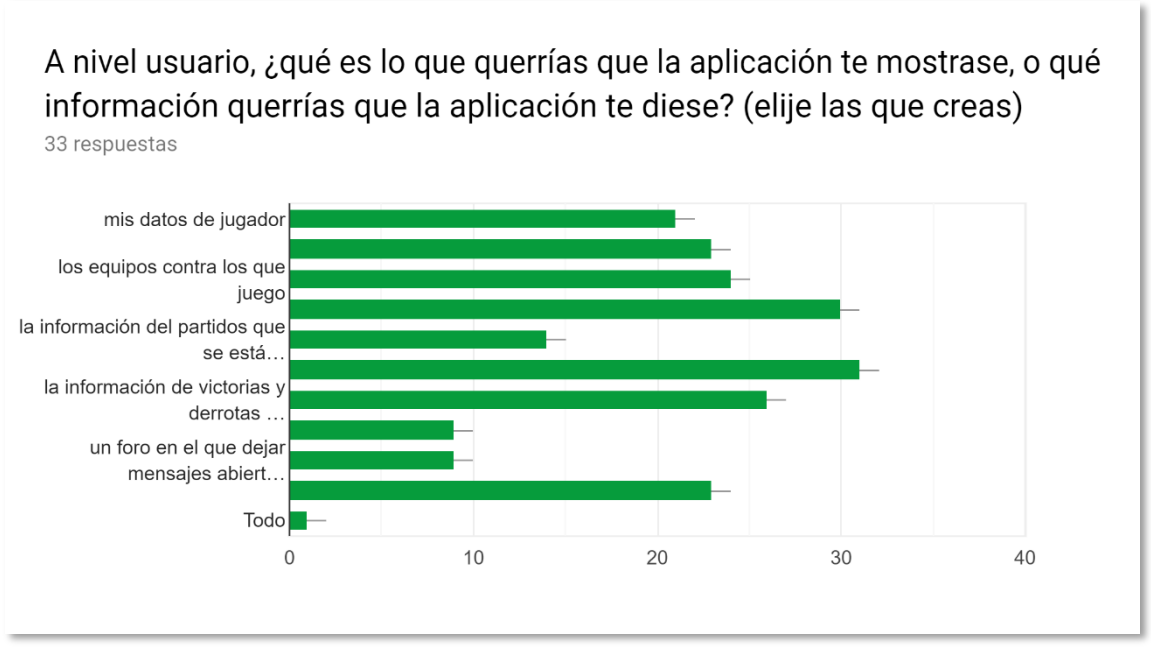
Aplicaciones de pádel que conocen	nº
Ninguna	26
Padel Manager	1
Todotorneos.com	1
Mis estadísticas	1
No contestan	4

La mayoría no conoce ninguna aplicación para gestionar ligas de cualquier deporte, a excepción de las analizadas "Padel Manager" y "TodoTorneos", como también "Mis estadísticas"



Aplicación que conocen	nº
Ninguna	27
Padel Manager	1
No contestan	5

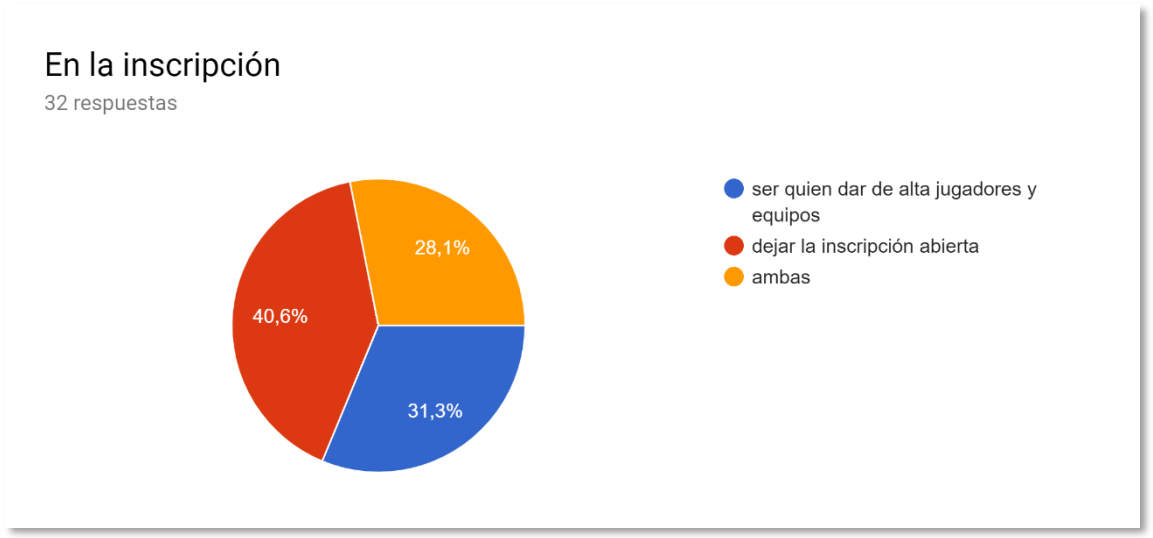
En este caso es evidente que los encuestados no conocen ningún tipo de aplicación para gestionar ligas de cualquier deporte, a excepción de un caso que conoce "Padel Manager"



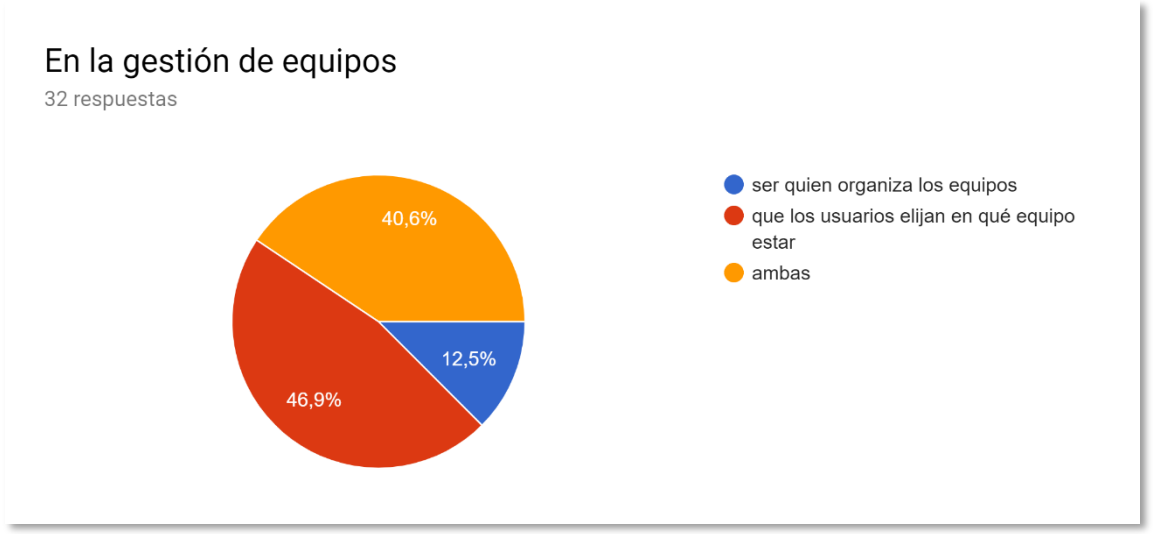
Organizada la información de manera más descriptiva, tenemos la siguiente tabla

Caracteristica de la aplicación	TOTAL
mis datos de jugador	22
los datos de los rivales contra los que juego	24
los equipos contra los que juego	25
el calendario con los próximos partidos	31
la información del partidos que se están jugando	15
los resultados de los partidos disputados	32
la información de victorias y derrotas y estadística de cada equipo y cada jugador	27
un chat para poder hablar entre los jugadores	10
un foro en el que dejar mensajes abiertos a todos los jugadores	10
poder configurar avisos para recordar partidos	24

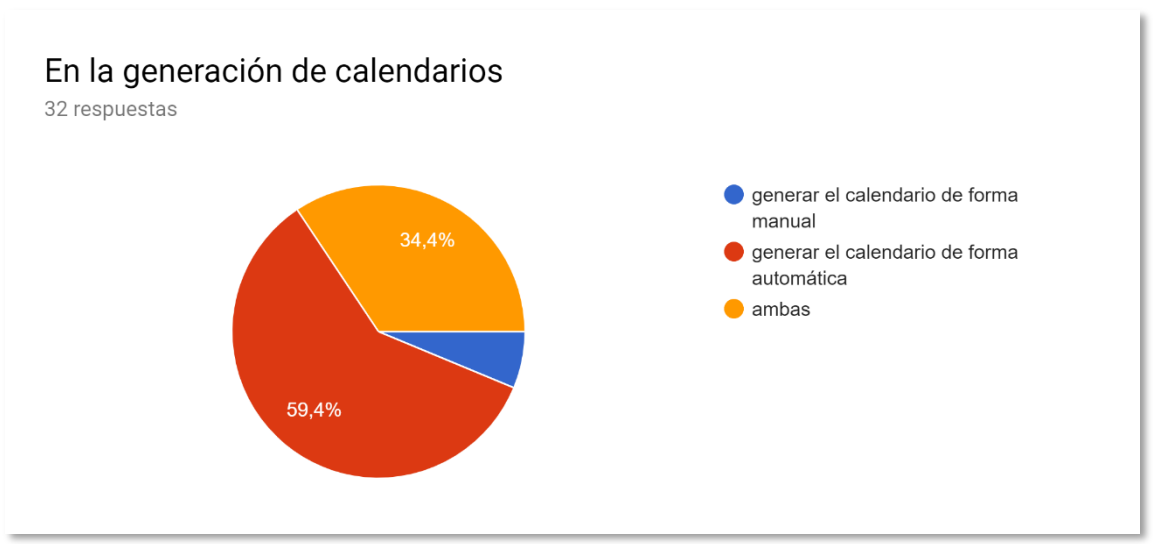
Siendo las opciones más destacadas las siguientes:
(32) Resultados de los partidos disputados
(31) Calendario con los próximos partidos
(27) Información de victorias y derrotas de cada equipo y jugador
(25) Equipos contra los que juego
(24) Datos de los rivales y (24) Avisos en la aplicación
(22) Mis datos de jugador



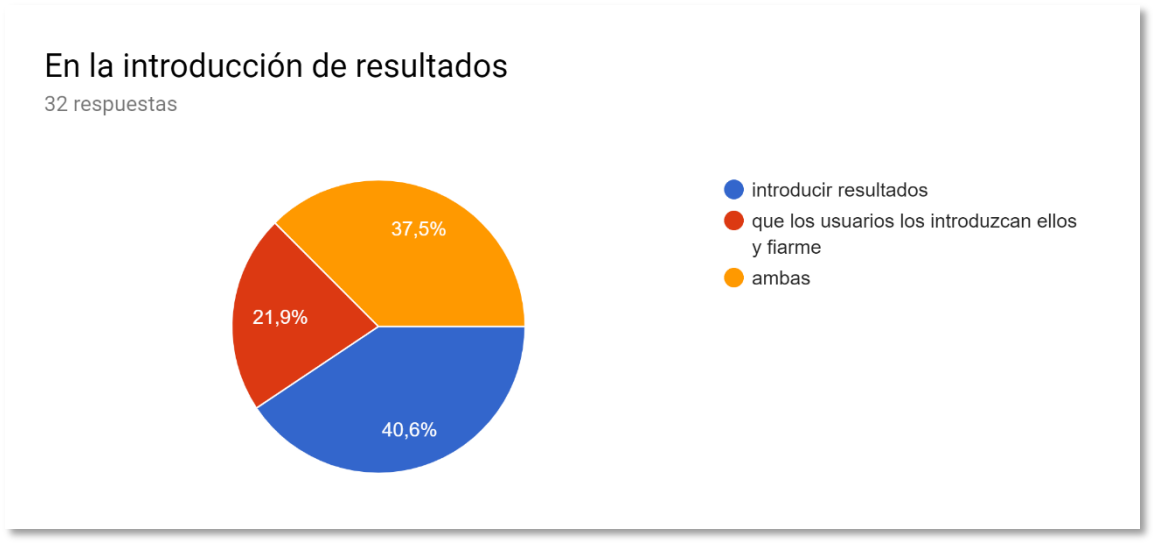
En relación a la inscripción, hay más porcentaje a favor de que los propios usuarios de la aplicación se inscriban en torneos que a controlar las inscripciones a nivel administrador, pero también es una opción bien valorada.

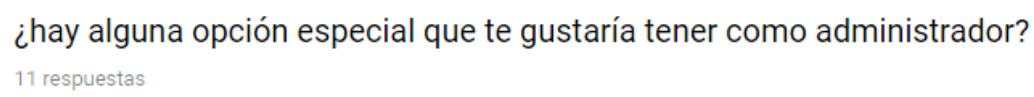


Como en el caso anterior, se prefiere dar la responsabilidad a los propios usuarios, pero con un control por parte de los administradores.



Una mayoría prefiere generar calendario de forma automática.



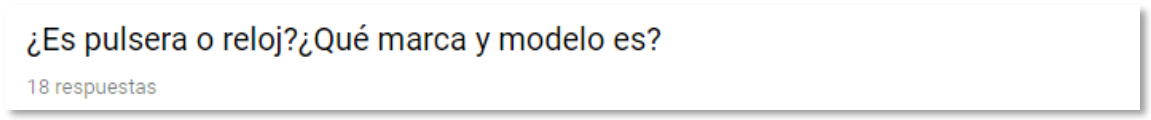


Descartando los comentarios sin valor añadido como los "no", tenemos estas 3 respuestas destacables, pues implica que entre 33 encuestados 4 han creído que su propuesta era tan buena como para indicarla.

- El resultado debe ser aceptado por ambos equipos
- Gestión de las pistas
- Una opción "broadcast" para enviar notificaciones a todos los usuarios podría ser útil, para informar cancelaciones, cambios, etc.
- Para la generación de calendarios, que el usuario pueda poner su disponibilidad y así no tener problemas con ese tema, en el caso que se tuviera que cancelar un partido se podría generar otro sabiendo ésa información.

Y para introducir los resultados, podríamos decir que prefieren que sean los administradores los que introduzcan estos, aunque también se valora que los jugadores los pongan.

7.6.6. Información sobre pulseras y relojes inteligentes



Los modelos que indican los encuestados son:

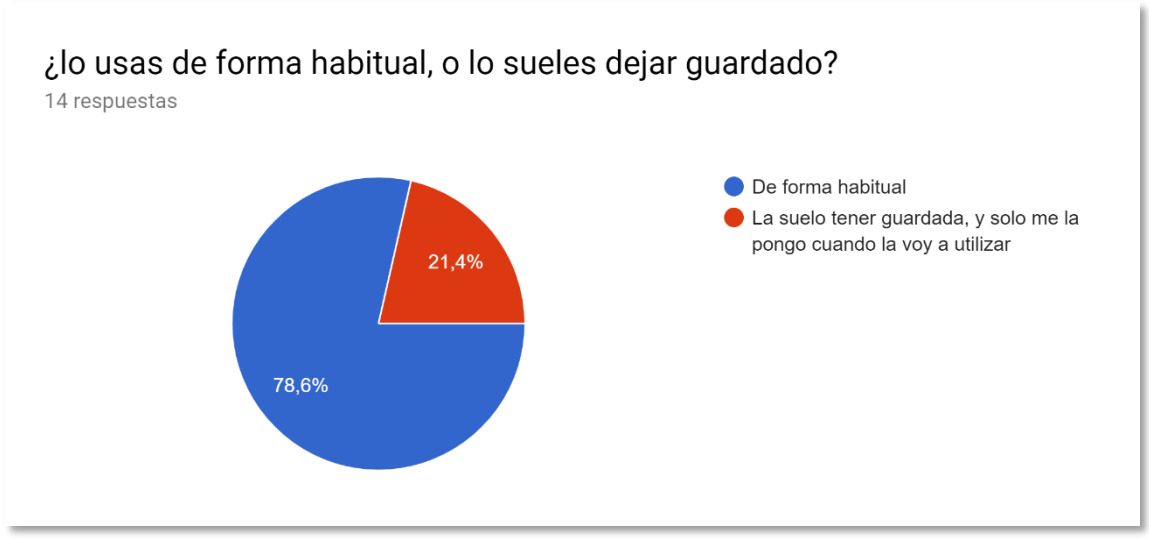
Marca y Modelo	nº
Reloj: Huawei watch 2	1
Pulsera: xaomi (mi band 2,3 y 4)	6
Reloj: Vaasco	1
Reloj: Samsung Galaxy Watch	1
Reloj: Garmin para running	1
Reloj: Garmin Vivo Active	1
Reloj: wilful	1
Reloj: mi3	1
Reloj: Polar m430	1
Reloj: xiaomi amazfit	1

De lo que sacamos en claro que la mayoría tienen relojes, y las pulseras se centran sobre todo en las del tipo "Mi band". Eso sí, dentro de los relojes no hay ninguno que destaque sobre el resto.

Tiempo	nº
4 años	1
3 años	2
2 años	4
1 año	4
1 dia	1
1 semana	1
2 meses	2

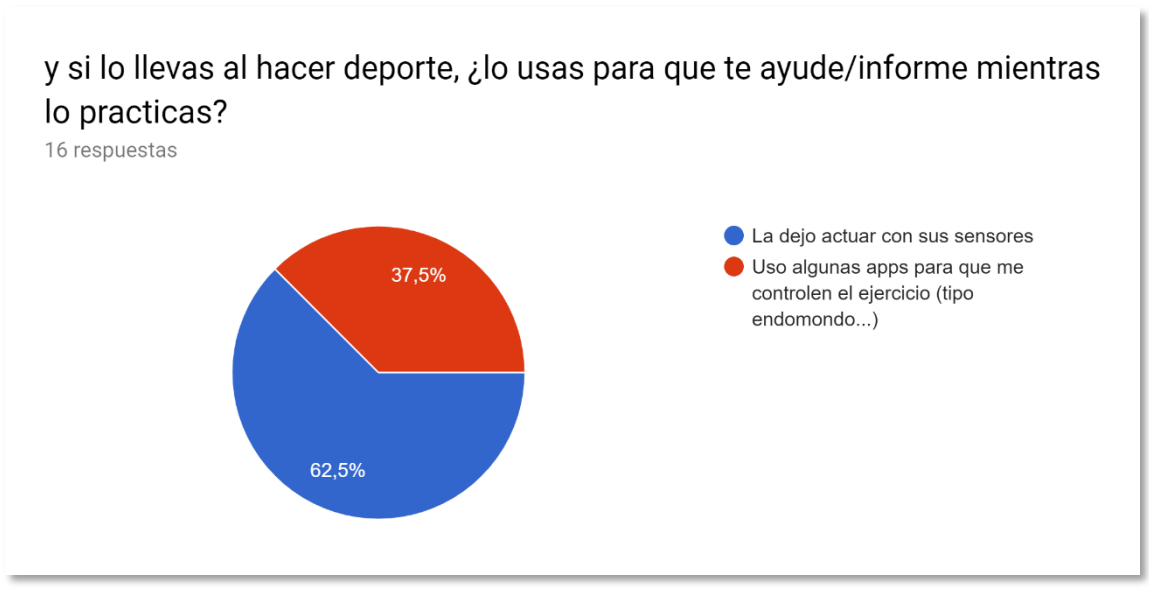
Tiempo	nº
1 año o más	11
menos de 1 año	4

Podemos sacar como conclusión que la información que den los encuestados, será valiosa, pues en su mayoría lo que usan este tipo de dispositivos tienen una experiencia de más de un año de uso, siendo en algunos casos más amplia.

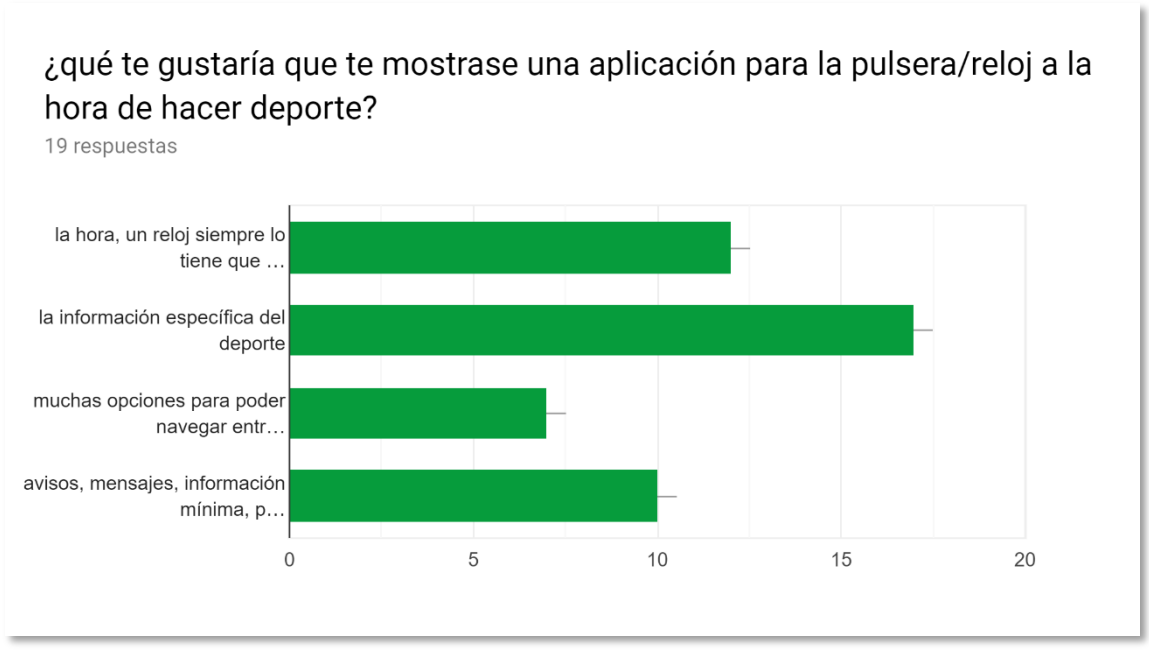


Casi un cuarto de los encuestados que tienen reloj o pulsera inteligente, reconocen dejarlo guardado y no usarlo todo el tiempo.

Y podríamos dar por casi mayoría, lo que hacen deporte usando su reloj.



Aunque no son mayoría los que usan aplicaciones específicas para controlar su deporte.



Organizado en una tabla

Caracteristica de la aplicación en el reloj	TOTAL
la información específica del deporte	17
la hora, un reloj siempre lo tiene que mostrar	12
avisos, mensajes, información mínima, poco más	10
muchas opciones para poder navegar entre ellas y elegir la que más me conviene	7

Podemos sacar como conclusión, que los usuarios de reloj inteligente prefieren, sobre todo, la información del deporte de forma concisa. Y la hora.