

Universidad de Alcalá

Escuela Politécnica Superior

Grado en Ingeniería Informática

Trabajo Fin de Grado

Desarrollo de una aplicación Android para Árbitros de Voleibol



ESCUELA POLITECNICA
SUPERIOR

Autor: Inés Viñuelas López

Tutor/es: Ana Castillo Martínez

2022

UNIVERSIDAD DE ALCALÁ
Escuela Politécnica Superior

Grado en Ingeniería Informática

Trabajo Fin de Grado
Desarrollo de una aplicación Android para Árbitros de
Voleibol

Autor: Inés Viñuelas López

Tutor/es: Ana Castillo Martínez

TRIBUNAL:

Presidente: << Nombre y Apellidos >>

Vocal 1º: << Nombre y Apellidos >>

Vocal 2º: << Nombre y Apellidos >>

FECHA: << Fecha de depósito >>

Agradecimientos

Quiero agradecer en mi primer lugar a mi familia, más concretamente a mis padres y a mi hermana, por apoyarme durante toda la carrera y motivarme a conseguir todas mis metas. Sin vuestro apoyo emocional nunca habría logrado llegar donde estoy hoy.

En segundo lugar quiero agradecer a mi pareja, por aguantarme en los peores momentos y estar a mi lado aún más en todos los buenos. Gracias por ayudarme a conseguir la carrera.

En tercer lugar quiero agradecer a mis amigos por estar a mi lado en todo momento, sobre todo a ti Isabel, por apoyarme durante tantos años, por llevar a cabo todos los planes juntas, por ser mi mano derecha y ayudarme en cualquier problema con la carrera. Sin ti nada hubiese sido lo mismo, gracias por hacerme ver la vida de una forma diferente. A ti también Pablo, por hacerme reír tanto, siempre tener una palabra buena hacia mí y ser un fiel compañero.

Por último, quiero agradecer a mi tutora del Trabajo de Fin de Grado, que ha tenido siempre un hueco para resolver todas mis dudas y por guiarme y acompañarme en el último paso de mi carrera.

Índice

1. Introducción	12
1.1. Motivación	13
2. Objetivo.....	14
3. Estado del arte	15
3.1. Teoría arbitral.....	15
3.1.1. Señales arbitrales.....	18
3.1.2. Sistemas de formación.....	21
3.2. Aplicaciones actuales	23
3.2.1. Volleyball Referee.....	23
3.2.2. AppLineup.....	24
3.2.3. E-Scoresheet.....	24
3.2.4. Data Video Essential	25
3.2.5. Data Volley	26
3.2.6. Comparativa entre aplicaciones	28
3.3. Entornos de desarrollo aplicaciones móviles	28
3.3.1. Java ME.....	29
3.3.2. Symbian.....	29
3.3.3. Android	29
3.3.4. Windows Mobile	29
3.3.5. Qt framework	29
3.3.6. BREW	30
3.3.7. Palm OS	30
3.4. Lenguajes de desarrollo de aplicaciones móviles	30
3.4.1. Desarrollo nativo Android	30
3.4.2. Desarrollo nativo en iOS	30
3.4.3. Desarrollo Híbrido.....	31
3.4.4. Apache Cordova.....	31
3.4.5. PhoneGap	32
3.4.6. Flutter	32
3.4.7. Ionic	32
3.4.8. Comparativa entre aplicaciones nativas e híbridas.....	33
3.5. Bases de datos	34
3.5.1. MySQL.....	34
3.5.2. PostgreSQL	34
3.5.3. MongoDB.....	35
3.5.4. MariaDB.....	35
3.5.1. Comparativa de bases de datos	35
3.6. Servidores Web	38
3.6.1. Apache Tomcat.....	38
3.6.2. Nginx.....	38
3.6.3. LiteSpeed.....	39
3.6.4. XAMPP.....	39
3.6.5. Insomnia.....	39
4. Desarrollo	41
4.1. Solución propuesta	41
4.2. Arquitectura del proyecto	41
4.2.1. Base de datos en PostgreSQL	42

4.2.1.	Servidor Web.....	58
4.2.2.	Aplicación servidor	59
4.2.3.	MQTT - Mosquitto	63
4.2.4.	Aplicación móvil	67
4.3.	Testimonios de árbitros	78
5.	Futuras mejoras	79
6.	Conclusiones	80
7.	Coste del proyecto	81
7.1.	Coste de personal.....	81
7.2.	Coste de material	82
7.3.	Costes Generales	82
7.4.	Coste Total	83
8.	Bibliografía.....	84
9.	Anexo I – Manual de usuario.....	86
10.	Anexo II – Glosario	91

Índice de Imágenes

Ilustración 1. Acta oficial completa.....	17
Ilustración 2. Acta sin completar.....	17
Ilustración 3. Sistema de juego 6-0.	22
Ilustración 4. Sistema de juego 4-2.	22
Ilustración 5. Formación en el sistema 5-1.	23
Ilustración 6. Imagen aplicación Volleyball Referee.....	24
Ilustración 7. Imagen aplicación AppLineup.....	24
Ilustración 8. Imagen Aplicación E-ScoreSheet.....	25
Ilustración 9. Imagen Aplicación Data Video Essential.....	26
Ilustración 10. Imagen Aplicación Data Volley.....	27
Ilustración 11. Comparativa apps híbridas y nativas.....	31
Ilustración 12. Diagrama de elementos.....	42
Ilustración 13. Diagrama de componentes.....	43
Ilustración 14. Interfaz Xampp.....	58
Ilustración 15. Imagen estructura servidor.	59
Ilustración 16. PreparedStatement InsertNewLogin.	63
Ilustración 17. Conexión MQTT.	64
Ilustración 18. Estructura Mosquitto.	65
Ilustración 19. Diagrama de pantallas de la aplicación.....	68
Ilustración 20. Código para establecer el tamaño de la columna.	71
Ilustración 21. Creación de las tablas, párrafos y celdas.....	71
Ilustración 22. Añadir imagen y tablas al PDF.	72
Ilustración 23. Código para la lectura del PDF.....	72
Ilustración 24. Código para escribir en el acta en PDF.....	72
Ilustración 25. Pasar información código QR.....	73
Ilustración 26. Llamada a la tarea TaskSelectPartidos.....	73
Ilustración 27. Insertar datos en la bbdd.	75
Ilustración 28. Publish MQTT.....	76
Ilustración 29. Suscripción a los topics.	76
Ilustración 30. Android Manifest.....	77
Ilustración 31. Gradle App.....	77
Ilustración 32. Coste del personal de secretaría.....	81
Ilustración 33. Coste total personal.	81
Ilustración 34. Imagen Login.	86
Ilustración 35. Imagen Registro.	86
Ilustración 36. Insertar rotación por parte del entrenador.	87
Ilustración 37. Imagen selección de datos.	87
Ilustración 38. Botón QR.	88
Ilustración 39. Información completa mediante código QR.	88
Ilustración 40. Establecer punto o sustitución.	88
Ilustración 41. Introducir el colocador en pista.....	89
Ilustración 42. Mensaje de error al solicitar más de un tiempo.....	89
Ilustración 43. Insertar sanción.....	89
Ilustración 44. Acta descargada en el dispositivo.....	90
Ilustración 45. Pantalla con el acta.....	90
Ilustración 46. Acta con datos.....	90
Ilustración 47. Imagen menú tres puntos.....	90

Índice de tablas

Tabla 1. Altura red Voleibol.....	15
Tabla 2. Señas de arbitraje de voleibol.....	21
Tabla 3. Comparación versiones aplicación Data Volley	27
Tabla 4. Comparativa de aplicaciones	28
Tabla 5. Tabla comparativa aplicaciones híbridas y nativas	33
Tabla 6. Comparativa de lenguajes de programación	33
Tabla 7. Comparativa entre bases de datos relacionales y no relacionales	37
Tabla 8. Comparativa entre MySQL y PostgreSQL	37
Tabla 9. Coste de personal de ingeniería.....	81
Tabla 10. Coste Hardware	82
Tabla 11. Coste Software	82
Tabla 12. Coste Software	82
Tabla 13. Costes Generales	83
Tabla 14. Coste Total	83

Resumen

En el proyecto expuesto en este documento, se ha creado una aplicación Android que favorece el trabajo de los árbitros de voleibol automatizando procesos realizados manualmente. Para lograr generar dicha aplicación, se ha creado una base de datos que almacena toda la información necesaria para llevar a cabo un partido. Adicionalmente, se ha generado un servidor que administra la base de datos y envía información a la aplicación para su posterior uso. A su vez, el servidor también recibe información de la aplicación para almacenarla en la base de datos, generando una actualización de los datos en tiempo real.

Palabras Clave

App, Referee, Minutes, Volleyball, Game.

Abstract

In the project described in this document, an Android application has been created to help volleyball referees in their work by automating processes carried out manually. To generate this application, a database has been created that stores all the information necessary to carry out a match. Additionally, a server has been generated that manages the database and sends information to the application for its subsequent use. In turn, the server also receives information from the application to store it in the database, generating an update of the data in real time.

Key words

App, Referee, Minutes, Volleyball, Game.

1. Introducción

Actualmente, la mayoría de las personas poseen un dispositivo inteligente mediante el cual se pueden instalar aplicaciones y gestionar el deporte desde un enfoque más tecnológico. De esta manera, el deporte es mucho más cercano para los usuarios, lo que supone explotar el 100% del potencial de los deportistas. Así mismo, los amantes del deporte han cambiado sus hábitos en el consumo del deporte. Casi cualquier deportista monitoriza sus sesiones de entrenamiento, sus constantes vitales o rutas a través de aplicaciones móviles [1].

La tecnología en el deporte [2] se inició en el mundo del motor, extendiéndose al resto de deportes como el fútbol, instalando el VAR para repetir jugadas, termografía para llevar la seguridad y protección de los deportistas o el ojo de halcón en el tenis para visualizar una acción dudosa.

Este proyecto se orienta al ámbito deportivo, más concretamente al voleibol, para ayudar en el trabajo de los árbitros de voleibol Nivel I mediante una aplicación Android que mecanice procedimientos arbitrales que hasta la fecha se realizan manualmente. Debido a que los árbitros Nivel I son novatos, surge la necesidad de realizar una simplificación de los procesos para agilizar los partidos y evitar errores humanos.

Durante los últimos años, todos los partidos arbitrados en este Nivel I se realizan de forma manual, ya que estos encuentros no suponen mucha exigencia a la hora del arbitraje y se evitan costes de presupuesto. Sin embargo, en la actualidad existen aplicaciones que mecanizan muchos procesos tales como digitalizar el acta o llevar un partido en tiempo real, reduciendo tiempos de ejecución y llevando un control por todos los árbitros durante el encuentro.

Por otro lado, estas aplicaciones están destinadas al colegiado más experto produciendo una carencia a la hora de sistematizar procesos más simples debido a que la trayectoria profesional y la formación de los árbitros a nivel nacional e internacional ha sido mucho más amplia.

Si por ejemplo un árbitro Nivel I quisiese usar estas aplicaciones, necesitaría realizar un curso previo suponiendo un gasto de dinero para obtener el título de acta electrónica.

En la actualidad existe una mayor cantidad de árbitros novatos debido a que suelen ser más jóvenes y no tienen tantas responsabilidades y trabajo extra como un árbitro nacional o internacional. Adicionalmente, existen más partidos a niveles amateur ya que no todos los jugadores tienen la motivación y el tiempo para aplicarlo a las ligas más profesionales.

En conclusión, es necesario crear una aplicación gratuita, fácil de usar, sin cursos previos, que refuerce las decisiones del colegiado y que recoja todas las peticiones y necesidades de los árbitros Nivel I agilizando los procesos arbitrales y que resuelva todas las dudas que puedan surgir durante los encuentros.

1.1. Motivación

Se va a llevar a cabo una aplicación Android que sistematice los procedimientos, ayude a los árbitros noveles a resolver sus dudas durante los encuentros e incluso añada funcionalidades nuevas para cubrir más necesidades.

Por tanto, esta aplicación está enfocada en el ámbito deportivo, más concretamente el voleibol, para árbitros novatos.

La elección de este deporte surgió debido a la experiencia de la autora del trabajo, la cual lleva practicando voleibol desde los ocho años a nivel regional, siendo a su vez árbitro de Nivel I desde hace 7 años. Esta experiencia hace que se tenga un contacto estrecho con árbitros jóvenes, para lo cual se demanda cada día más el uso de aplicaciones fáciles y gratuitas que resuelvan y sistematicen los procedimientos arbitrales.

Es por ello por lo que se considera que es necesario crear una aplicación que ayude a resolver y apoyar las decisiones arbitrales dando como resultados partidos más ágiles, limpios y sin errores humanos.

2. Objetivo

El objetivo principal de este proyecto es el desarrollo de una aplicación Android que permita agilizar los encuentros de voleibol automatizando procesos arbitrales que se realizan manualmente. Esta aplicación estará destinada principalmente a árbitros noveles. Esta aplicación está destinada al colegiado y entrenadores de Voleibol, por tanto, su uso está centrado solo a este deporte.

Para realizar dicho objetivo, se debe implementar un sistema que realice las actas de los partidos digitalmente en tiempo real incluyendo una base de datos con los jugadores. A su vez, implantar códigos QR para leer las hojas de rotación.

3. Estado del arte

El mundo del deporte ha avanzado mucho gracias a la tecnología actual. Actualmente, existen aplicaciones para monitorizar pulsaciones, grabar recorridos e incluso acortar la recuperación de lesiones. Por ejemplo, en el tenis se ha diseñado el ojo de halcón donde se pueden observar jugadas dudosas o en el fútbol se ha creado el VAR para repetir jugadas mediante video arbitraje [1].

De dar a aplicar correctamente la tecnología disponible es necesario realizar un primer estudio donde se recojan las necesidades de actuación de los árbitros, ya que se trata del público objetivo del trabajo. Seguidamente se analizan las aplicaciones existentes que servirán de referencia para la realización del proyecto. Una vez analizadas estas aplicaciones se analizarán los elementos necesarios para realizar el desarrollo tales como entornos de desarrollo, bases de datos, etc.

3.1. Teoría arbitral

Para entender el funcionamiento que es necesario desarrollar en la aplicación, así como la terminología utilizada en ella, es necesario realizar un estudio a la teoría arbitral y normativa de voleibol [3].

Dentro del área de juego, el primer árbitro debe comprobar que la red esta correctamente tensada. A su vez, el segundo árbitro debe medir la altura de la red antes del sorteo mediante una vara de medición (si es posible, metálica) diseñada para este propósito. En la vara se deben marcar las alturas dependiendo de su categoría para hombres y mujeres respectivamente. El 1er árbitro permanece cerca del 2do árbitro durante esta verificación para supervisar y confirmar la medición.

Categoría	Masculina	Femenina
Senior	2,43	2,24
Juvenil	2,43	2,24
Cadete	2,37	2,18
Infantil	2,24	2,10
Alevín	2,10	2,10
Benjamín	2,00	2,00

Tabla 1. Altura red Voleibol

Respecto al balón, El segundo árbitro, antes del comienzo del juego, toma posesión de los 5 balones que se utilizarán en el partido y comprueba que todos tengan características idénticas (color, circunferencia, peso y presión siendo el responsable de los balones durante todo el partido.

En relación con los equipos, la composición de un equipo se determina en el Reglamento Específico de la Competencia, y puede consistir de hasta 17 personas, que comprenden hasta 12 jugadores, entre ellos hasta 2 jugadores Libero, y hasta 5 oficiales. Los 5 oficiales permitidos en la banca son elegidos por el entrenador, siempre que estén listados en el O2-bis. Para

Competencias Mundiales y Oficiales de la FIVB de Mayores, hasta 14 jugadores pueden ser registrados en la hoja de anotación, y jugar en el partido.

El entrenador y el capitán del equipo (ambos verifican y firman la hoja de anotación, o la lista del equipo en la hoja de anotación electrónica) son responsables de la identidad de los jugadores registrados en la lista de la hoja de anotación. Una consideración muy importante a tener en cuenta es la revisión de los dorsales de los jugadores en pista con los que se han registrado en la hoja de rotación, evitando las posibles discrepancias que pudiese haber.

En cuanto a la estructura del juego, al final de cada set, el segundo árbitro inmediatamente debe pedirles a los entrenadores las hojas de rotación con las posiciones de los jugadores en campo. En el caso de que una vez se saque, los jugadores del equipo contrario estén mal posicionados dependiendo del sistema de formación que tengan, el segundo arbitro debe indicar dicha falta.

Por último, el anotador debe inscribir en el acta todos los datos del partido, incluyendo los jugadores y entrenadores, comprobando los documentos oficiales de acreditación con la hoja federativa, en caso de algún error, el anotador deberá reflejarlo en el apartado de observaciones del acta, tras esto, deberá pedir la firma a los entrenadores acreditando que están de acuerdo con los jugadores inscritos en el acta. Una vez iniciado el partido, el anotador debe estar pendiente de las posibles sustituciones que puedan realizarse, anotándolas en el acta, así como del tanteo, los tiempos muertos, las sanciones y la rotación de cada set, comprobando los números en pista con los inscritos en el acta. Una vez finalizado el partido, deberá completar el recuadro final de puntuación y pedir la firma de aprobación de los capitanes.

Como posibles casos para tener en cuenta, el anotador debe escribir en el lado izquierdo del acta al equipo local y en el lado derecho al visitante. A su vez, el primer apellido de un jugador debe corresponder con el capitán de cada equipo. En el caso de que haya un jugador inscrito que sea menor de la categoría que se disputa, se deberá reflejar el último en el acta y entre paréntesis la categoría que le corresponde. Adicionalmente, si hay dos jugadores con el mismo apellido, se inscribe una coma seguida de la inicial del segundo apellido, en el caso de ser hermanos, se escribe el apellido seguido de un punto y la primera inicial del nombre. En el recuadro de los líberos, primero se inscribe al libero principal y después al secundario en caso de que aplique. Si no se ha completado toda la tabla de jugadores, deberá cerrarse el acta con una Z. A continuación, se muestra un ejemplo de un acta oficial completada y de un acta sin completar:

Nombre de la Competición: **LIGA CADETE REGIONAL**

Ciudad: **GUADALAJARA** Código: Fecha: **15.01.2022** Hora: **11:30**

Cancha: **DON BOSCO** Partido Nº: **1**

División: **Masc. Femen. X** Categoría: **Sen. Juv. Cad. Int. Ale.** **Local** EQUIPOS **Visitante** **REAL FEDERACION ESPAÑOLA DE VOLEIBOL ACTA DE PARTIDO**

Formación equipo: **1** Orden al saque: **SAL (A)** N° jugadores iniciales: **6** N° jugador: **19 7 5 16 11 9** Sustitución: **1 2 3 4 5 6** Tanteo al cambio: **1 2 3 4 5 6** Turnos al saque: **1 2 3 4 5 6**

Formación equipo: **2** Orden al saque: **TON (B)** N° jugadores iniciales: **6** N° jugador: **7 5 3 6 18 8** Sustitución: **1 2 3 4 5 6** Tanteo al cambio: **1 2 3 4 5 6** Turnos al saque: **1 2 3 4 5 6**

Formación equipo: **3** Orden al saque: **SAL (A)** N° jugadores iniciales: **6** N° jugador: **16 7 9 19 5 11** Sustitución: **1 2 3 4 5 6** Tanteo al cambio: **1 2 3 4 5 6** Turnos al saque: **1 2 3 4 5 6**

Formación equipo: **4** Orden al saque: **TON (B)** N° jugadores iniciales: **6** N° jugador: **7 5 3 6 18 8** Sustitución: **1 2 3 4 5 6** Tanteo al cambio: **1 2 3 4 5 6** Turnos al saque: **1 2 3 4 5 6**

Formación equipo: **5** Orden al saque: **SAL (A)** N° jugadores iniciales: **6** N° jugador: **16 7 9 19 5 11** Sustitución: **1 2 3 4 5 6** Tanteo al cambio: **1 2 3 4 5 6** Turnos al saque: **1 2 3 4 5 6**

SANCCIONES

OBSERVACIONES

APROBACION

Arbitros: **1ª VIRUELAS, INES** Licencia: **15253** Firma: **[Firma]**

Asist. **1** Jueces de Línea **2**

RESULTADO FINAL

EQUIPO	S	P	SET	PUNTO	EQUIPO	S	P	SET	PUNTO
SAL (A)	0	1	25	1	TON (B)	3	0	0	0
SAL (A)	0	1	25	1	TON (B)	3	0	0	0
SAL (A)	0	2	12	1	TON (B)	3	0	0	0
SAL (A)	0	2	12	3	TON (B)	3	0	0	0
SAL (A)	0	2	12	3	TON (B)	3	0	0	0
SAL (A)	0	2	12	3	TON (B)	3	0	0	0

JUGADOR LIBERO ("L")

OFICIALES

FIRMAS

Capitán: **[Firma]** Capitán: **[Firma]**

Entrenador: **[Firma]** Entrenador: **[Firma]**

Ilustración 1. Acta oficial completa

Nombre de la Competición: **Local** EQUIPOS **Visitante** **REAL FEDERACION ESPAÑOLA DE VOLEIBOL ACTA DE PARTIDO**

Ciudad: **Local** Código: Fecha: **Local** Hora: **Local**

Cancha: **Local** Partido Nº: **Local**

División: **Masc. Femen. X** Categoría: **Sen. Juv. Cad. Int. Ale.** **Local** EQUIPOS **Visitante** **REAL FEDERACION ESPAÑOLA DE VOLEIBOL ACTA DE PARTIDO**

Formación equipo: **1** Orden al saque: **SAL (A)** N° jugadores iniciales: **6** N° jugador: **19 7 5 16 11 9** Sustitución: **1 2 3 4 5 6** Tanteo al cambio: **1 2 3 4 5 6** Turnos al saque: **1 2 3 4 5 6**

Formación equipo: **2** Orden al saque: **TON (B)** N° jugadores iniciales: **6** N° jugador: **7 5 3 6 18 8** Sustitución: **1 2 3 4 5 6** Tanteo al cambio: **1 2 3 4 5 6** Turnos al saque: **1 2 3 4 5 6**

Formación equipo: **3** Orden al saque: **SAL (A)** N° jugadores iniciales: **6** N° jugador: **16 7 9 19 5 11** Sustitución: **1 2 3 4 5 6** Tanteo al cambio: **1 2 3 4 5 6** Turnos al saque: **1 2 3 4 5 6**

Formación equipo: **4** Orden al saque: **TON (B)** N° jugadores iniciales: **6** N° jugador: **7 5 3 6 18 8** Sustitución: **1 2 3 4 5 6** Tanteo al cambio: **1 2 3 4 5 6** Turnos al saque: **1 2 3 4 5 6**

Formación equipo: **5** Orden al saque: **SAL (A)** N° jugadores iniciales: **6** N° jugador: **16 7 9 19 5 11** Sustitución: **1 2 3 4 5 6** Tanteo al cambio: **1 2 3 4 5 6** Turnos al saque: **1 2 3 4 5 6**

SANCCIONES

OBSERVACIONES

APROBACION

Arbitros: **1ª** Nombre: **[]** Licencia: **[]** Firma: **[]**

Asist. **1** Jueces de Línea **2**

RESULTADO FINAL

EQUIPO	S	P	SET	PUNTO	EQUIPO	S	P	SET	PUNTO
SAL (A)	1	()			TON (B)	2	()		
SAL (A)	2	()			TON (B)	3	()		
SAL (A)	3	()			TON (B)	4	()		
SAL (A)	4	()			TON (B)	5	()		
SAL (A)	5	()			TON (B)	6	()		

JUGADOR LIBERO ("L")

OFICIALES

FIRMAS

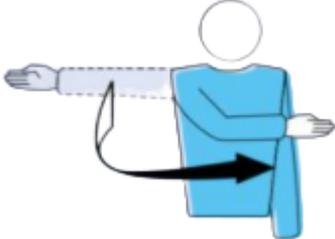
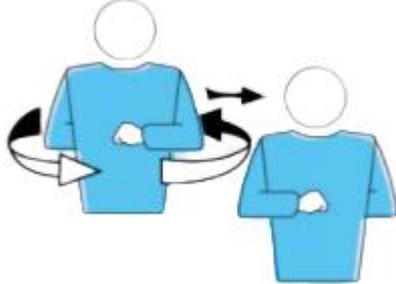
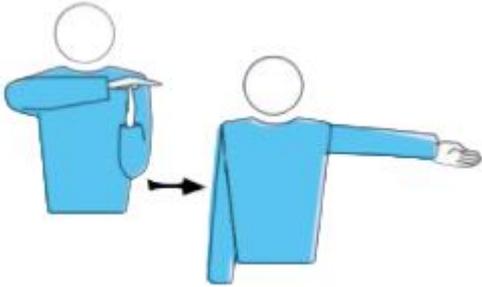
Capitán: **1 líbero** Capitán: **2 líbero**

Entrenador: **[]** Entrenador: **[]**

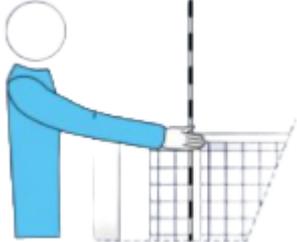
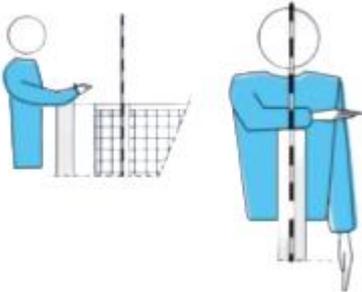
Ilustración 2. Acta sin completar

3.1.1. Señales arbitrales

Los árbitros deben señalar en cada momento la acción cometida, por ello, a continuación se detallan todas las posibles señas [4] que se pueden llevar a cabo durante un encuentro.

Regla	Descripción	Gesto
Autorización para el saque	Mover la mano para indicar la dirección del saque.	 <p>Un árbitro con un brazo extendido horizontalmente hacia la izquierda, con una flecha curva que indica el movimiento de la mano hacia la derecha.</p>
Equipo que saca	Extender el brazo hacia el lado del equipo que sacará.	 <p>Un árbitro con el brazo extendido horizontalmente hacia la derecha.</p>
Cambio de campo	Levantar los antebrazos sobre el pecho y espalda y girarlos alrededor del cuerpo.	 <p>Un árbitro con los antebrazos levantados sobre el pecho y la espalda, con flechas que indican el movimiento circular de los brazos.</p>
Tiempo muerto	Colocar la palma de una mano sobre los dedos de otra, mantener verticalmente (formando una T) e indicar el equipo solicitante.	 <p>Un árbitro con una mano sobre los dedos de la otra, formando una T, con una flecha que indica el movimiento hacia el lado del equipo solicitante.</p>
Sustitución	Movimiento circular de los antebrazos uno alrededor del otro.	 <p>Un árbitro con los antebrazos levantados y girados uno alrededor del otro.</p>

Regla	Descripción	Gesto
Amonestación por conducta incorrecta	Mostrar una tarjeta amarilla para amonestar.	
Castigo por conducta incorrecta	Mostrar una tarjeta roja para el castigo.	
Expulsión	Mostrar ambas tarjetas juntas para la expulsión.	
Retención	Levantar la mano con el antebrazo y la palma hacia arriba.	
Doble golpe	Levantar los dos dedos abiertos y levantados.	

Regla	Descripción	Gesto
Cuatro toques	Levantar cuatro dedos abiertos y levantados.	
Toque de red por jugador	Indicar el lado de la red con la mano correspondiente	
Invasión	Levantar el antebrazo plano y por encima de la red.	
Zaguero	Hacer un movimiento con la mano abierta.	
Penetración en el campo contrario	Indicar la línea central.	

Regla	Descripción	Gesto
Repetición de jugada	Levantar ambos pulgares hacia arriba.	
Balón tocado	Cepillar con la palma de la mano los dedos de la otra verticalmente.	
Amonestación por demora	Cubrir la muñeca con la tarjeta que corresponda.	

Tabla 2. Señas de arbitraje de voleibol.

3.1.2. Sistemas de formación

Se trata de una serie de movimientos tácticos [5] tras pasar una vez el balón al lado contrario, donde cada jugador se desplaza a una posición determinada del campo.

3.1.2.1. Sistema 6-0

Este sistema se caracteriza principalmente porque los jugadores no tienen un rol específico, es decir, todos tienen la misma posibilidad de hacer de colocador, atacante y defensor. El árbitro no tiene que prestar especial atención a la rotación.

En este caso el colocador será por turno, aquel jugador que pase por la posición previamente acordada será el que cumpla el rol de colocador.

Mayormente se selecciona la posición número 3 para determinar el jugador (colocador) que proporcionará la pelota para realizar alguna jugada.

Habitualmente se trata de un sistema donde reciben tres jugadores en forma de U. A continuación, se muestra una imagen representativa de dicha formación:

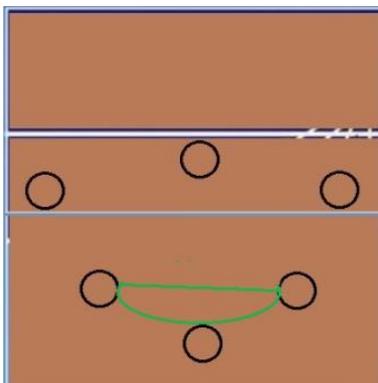


Ilustración 3. Sistema de juego 6-0.

3.1.2.2. Sistema 4-2

Este es un sistema básico-avanzado, se emplea con frecuencia ya que su lógica es sencilla de comprender, tiene una estructura bien específica.

Se denomina 4-2 haciendo referencia al número de jugadores designados como atacantes y colocadores. Al igual que el anterior, el segundo árbitro no tiene que prestar especial atención ya que los jugadores se colocan de una manera clara y sencilla de visualizar.

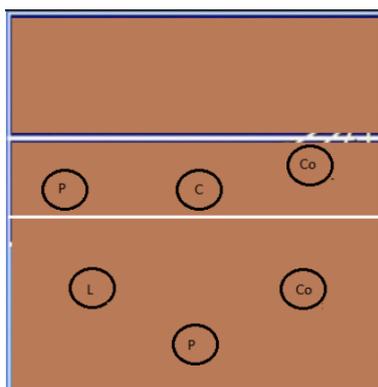


Ilustración 4. Sistema de juego 4-2.

3.1.2.3. Sistema 5-1

Este sistema tiene la posibilidad de 5 rematadores y un solo colocador que se ocupará de la construcción de las jugadas de ataque del equipo.

Tal sistema de juego en el voleibol se encuentra en lo más alto del nivel de especialización de todos los sistemas utilizados a nivel profesional.

Este sistema es el que produce problemas entre los árbitros noveles, ya que la colocación en el campo de los jugadores a la hora de la recepción es diferente, llevando a confusiones en el caso de que los jugadores se hayan colocado mal. El segundo arbitro debe prestar especial atención a la posición del colocador para encontrar las faltas de posición. A continuación, se muestran las diferentes rotaciones dependiendo de la posición del colocador:

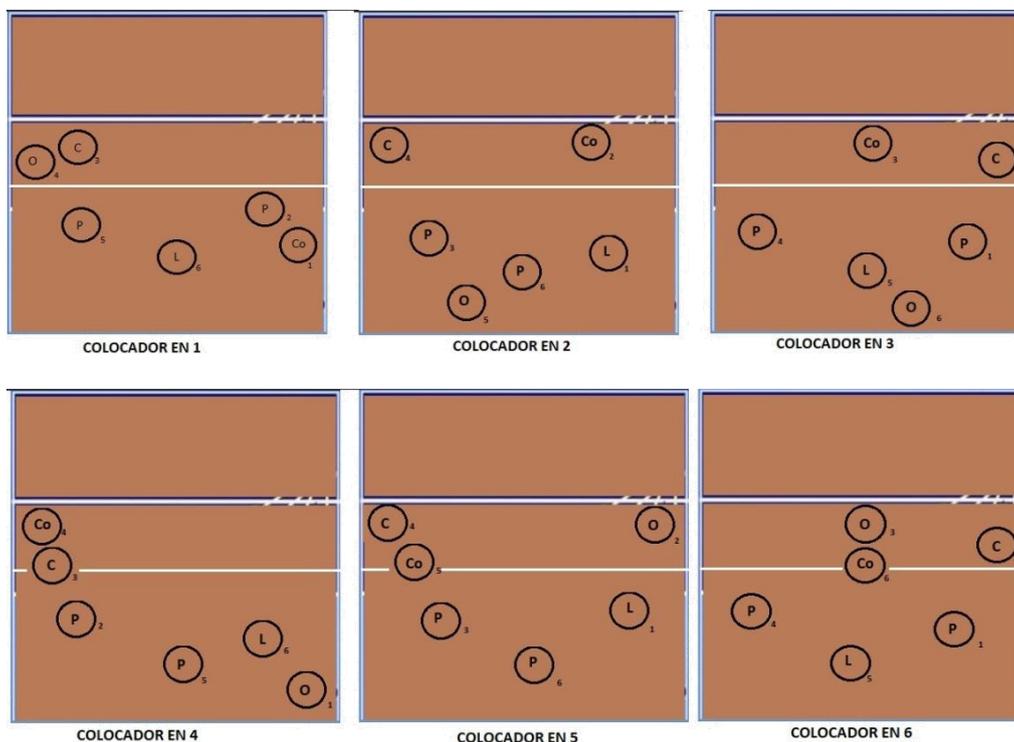


Ilustración 5. Formación en el sistema 5-1.

3.2. Aplicaciones actuales

En la actualidad existen muchas aplicaciones en el mercado para árbitros de voleibol que se usan tanto a nivel nacional e internacional. A continuación, se expondrán algunas de las aplicaciones más usadas y que son altamente competitivas.

3.2.1. Volleyball Referee

Volleyball Referee [6] es una aplicación completa y fácil de usar para arbitrar partidos de voleibol y vóley playa desarrollada por GuillaumeVT.

Esta aplicación, crea partidos de voleibol usando reglas oficiales o personalizadas por el usuario. A su vez, controla la puntuación, las rotaciones, los tiempos muertos e incluso el saque. Se pueden realizar sustituciones fácilmente y realizar el sorteo inicial. Se puede compartir los partidos en vivo por internet y se pueden guardar los encuentros en el dispositivo local. El acta generada en cada uno de los encuentros es posible guardarlo como PDF en el dispositivo local.

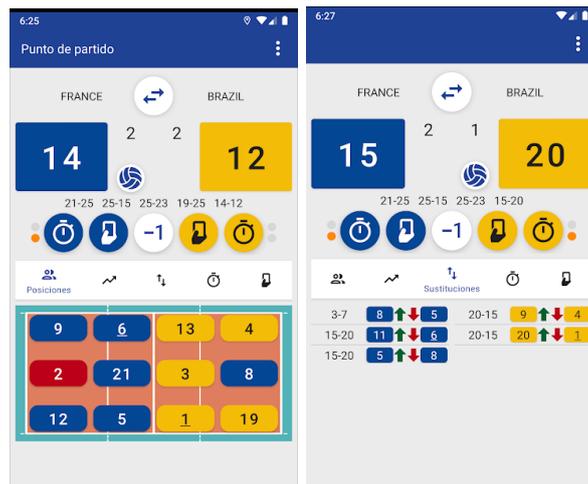


Ilustración 6. Imagen aplicación Volleyball Referee

3.2.2. AppLineup

AppLineup [7] es una aplicación usada para gestionar las hojas de rotación desarrollada por la empresa Veiem 360 SL. La idea de crear esta aplicación surgió a raíz del Covid-19, para evitar el contacto innecesario entre entrenadores y árbitros [8]. La aplicación genera un código QR a partir de la rotación introducida por los entrenadores correspondiente a cada set. El árbitro lee el QR y se almacena la información de todos los sets.



Ilustración 7. Imagen aplicación AppLineup

3.2.3. E-Scoresheet

E-Scoresheet [9] es una aplicación de escritorio que guía para compilar el acta del partido en tiempo real, a través de automatismos, con el fin de limitar al máximo la intervención del operador, mediante la comprobación de todas las situaciones e imprevistos que pueden ocurrir durante un partido y sugiriendo una solución eventual.

Permite la transmisión en tiempo real del marcador a la federación, permite interactuar con la base de datos de la Federación Internacional de Voleibol y usa aplicaciones externas como Data Video Essential, Data Volley, Volleyball e LiteScore, Scoresheets Referee, TV Score, explicadas posteriormente.

A partir de la temporada 2012/2013, es el único documento oficial que determina la validez del juego para todas las competiciones de la CEV y la FIVB. En la actualidad, muchas otras federaciones de todo el mundo utilizan la e-Scoresheet y muchas más y la hoja de resultados en papel al mismo tiempo.

Pero esta aplicación no es gratuita, por lo que supone un coste para la federación.



Ilustración 8. Imagen Aplicación E-ScoreSheet

3.2.4. Data Video Essential

La aplicación Data Video Essential [10] captura o importa el vídeo del partido en el PC, establece la posición del saque y del colocador en la pista, y con sólo 3 botones estarás listo para el *scouting*. Si tienes Click&Scout o Data Volley, puedes sincronizar automáticamente el vídeo y los datos estadísticos, empezando por el primer saque, para asociar cada golpe realizado al momento correspondiente del vídeo.

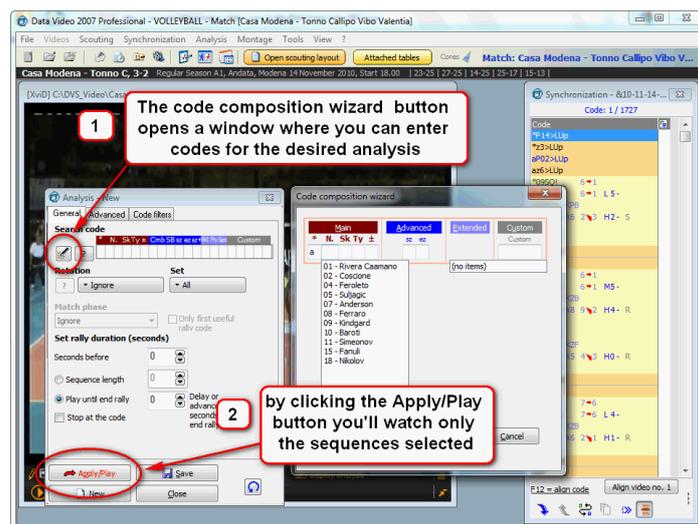


Ilustración 9. Imagen Aplicación Data Video Essential

3.2.5. Data Volley

Data Volley [11] es un software de estadísticas y análisis que permite a SeTTEX tener acceso en tiempo real a los datos recogidos por el *scoutman* para proporcionar al usuario una imagen automática, inmediata y realista de lo que está sucediendo en la cancha sin necesidad de llevar un registro en pantalla de dónde se toca la pelota.

Data Volley 4 Professional ofrece muchas funciones adicionales para el personal durante y después del partido. La App Cliente Web permite comunicarse con el banquillo y los medios de comunicación, el Cliente para obtener un análisis más preciso. El Reader permite utilizar las funciones de Data Volley 4 incluso sin la llave, mientras que con la Player App se pueden volver a ver diferentes situaciones de juego.

Data Volley 4 sólo funciona con el sistema operativo Windows. Para funcionar sin ralentizaciones, Data Volley 4 necesita un ordenador con al menos 4 GB de RAM, un puerto USB "Tipo A" para insertar el dongle y una conexión a Internet para la activación de la licencia y la descarga automática de las actualizaciones. Data Volley 4 está disponible en italiano, inglés, español (castellano y latinoamericano), japonés, ruso, alemán, francés, polaco, holandés, portugués y chino.

Esta aplicación también es de pago, rodando su precio en los 300 euros al año la versión Lite y en 800 euros al año la versión completa.

Data Volley	Profesional	Lite
Lo mejor para	PRO LEVEL	MEDIUM LEVEL
Precio	Annual License €799	Annual License €299
VIDEO BREAKDOWN	Import and synchronize	Import and synchronize
	Capture - Streaming	-

Data Volley	Profesional	Lite
DATA GATHERING	Offline - Online Calendar	Offline - Online Calendar
	Both teams	Both teams
	Export scouting format dv4, 2007	Export scouting format dv4, 2007
	Export scouting format dv2 - vis	-
	Extended code - Custom code	-
DATA ANALYSIS	Player - Skill - Rotation	Player - Skill - Rotation
	Zone - Direction	Zone - Direction
	General stats - Point	General stats - Point
	Combination Analysis	Combination Analysis
	Setter calls distribution	Setter calls distribution
	Advanced filters - Advanced analysis	-
VIDEO ANALYSIS	Free search	Free search
	Statistics - Rotations Search	Statistics - Rotations Search
	Montage	-
	Advanced search	-
LIVE AND POST-MATCH CLIENT	Web Client App	-
	Client	-
	Reader	-
	Player App	-

Tabla 3. Comparación versiones aplicación Data Volley

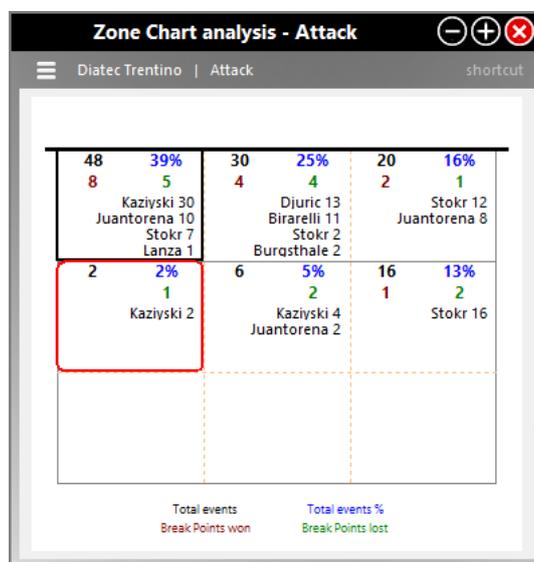


Ilustración 10. Imagen Aplicación Data Volley

3.2.6. Comparativa entre aplicaciones

En este apartado se realiza una comparativa entre todas las aplicaciones existentes en el mercado con la aplicación diseñada, mostrando las principales características de cada una de ellas.

	Volleyball Referee	AppLineup	E-Score Sheet	Data Video Essential	Data Volley	Play Volleyball (Desarrollada en este proyecto)
Tiene hojas de rotación	No	Sí	No	No	No	Sí
Hojas de rotación con QR	No	Sí	No	No	No	Sí
Sistema de juego en tiempo real	Sí	No	Sí	No	No	Sí
Manejo de tiempos muertos	Sí	No	Sí	No	No	Sí
Manejo de sustituciones	Sí	No	Sí	No	No	Sí
Manejo de sanciones	Sí	No	Sí	No	No	Sí
Faltas de rotación	No	No	No	No	No	Sí
Acta en tiempo real	Sí	No	Sí	No	No	Sí
Acta en PDF	Sí	No	Sí	No	No	Sí
Uso de BBDD	No	No	Sí	No	No	Sí
Estadísticas	No	No	Sí	Sí	Sí	No
Multiusuario	No	No	Sí	No	Sí	Sí
Video	No	No	Sí	Sí		No
Gratis	Sí	Sí	No	No	No	Sí
Disponible	Android y Apple	Android y Apple	Android y Apple	Windows	Windows	Android

Tabla 4. Comparativa de aplicaciones

3.3. Entornos de desarrollo aplicaciones móviles

El desarrollo de aplicaciones móviles [12] es el conjunto de procesos y procedimientos involucrados en la escritura de software para dispositivos informáticos pequeños e inalámbricos, como teléfonos inteligentes y otros dispositivos portátiles.

3.3.1. Java ME

Java ME es una edición de Java orientada a dispositivos pequeños consistiendo en una máquina virtual conjunto del API5. Esta plataforma produce normalmente aplicaciones portables, aunque algunas veces existen librerías específicas de cada dispositivo (comúnmente usadas para juegos), que las hacen no portables. Las aplicaciones de Java ME se pueden emular en un PC durante la fase de desarrollo y luego se pueden cargar fácilmente las aplicaciones en el móvil. Aunque el proceso no sea directo, resulta bastante económico portarlas a otros dispositivos al utilizar tecnologías Java para el desarrollo.

3.3.2. Symbian

La Plataforma Symbian es un sistema operativo de tiempo real, multitarea, específicamente pensada en funcionar bien en sistemas con recursos limitados, así como para maximizar la eficiencia y la vida de la batería y minimizar, de esta manera, el uso de memoria. La Symbian Foundation mantiene el código para la plataforma de software libre basada en Symbian OS y las aportaciones de software de Nokia, NTT DOCOMO y Sony Ericsson, e incluye las interfaces de usuario S60 y MOAP(S). La plataforma es de código abierto en su totalidad, y la mayoría se proporciona bajo la Licencia Pública de Eclipse.

3.3.3. Android

Android ha sido la plataforma elegida para el desarrollo de la aplicación. Se ha elegido porque es una plataforma basada en Linux de la Open Handset Alliance, entre cuyos treinta y cuatro miembros se encuentran Google, HTC, Motorola, Qualcomm y T-Mobile. Por lo tanto, treinta y cuatro de las principales compañías de software, hardware y telecomunicaciones dan soporte a esta plataforma. La programación de aplicaciones se hace básicamente en Java aunque se puede realizar mediante Kotlin, por tanto su programación es muy sencilla.

3.3.4. Windows Mobile

Windows Mobile es una variante de Windows CE para teléfonos móviles. Originalmente, Windows CE se desarrolló para ordenadores de bolsillo y PDA con pantallas táctiles que funcionaban con un stylus, y se adaptó posteriormente para su uso en smartphones equipados con un teclado. Este entorno soporta un subconjunto de la interfaz de programación de Win32 y una GUI13 simplificada con una ventana en la pantalla a la vez.

3.3.5. Qt framework

Qt usa estándar C++, pero hace un uso extensivo de un preprocesador especial llamado MOC¹⁴¹ para enriquecer el lenguaje. También se puede usar Qt en otros lenguajes de programación

¹ Meta object compiler

utilizando enlaces entre lenguajes. Funciona sobre las principales plataformas y tiene un soporte internacional extenso. Entre las características no relacionadas con la GUI, se encuentra el acceso a bases de datos SQL, el tratamiento de XML, la gestión de threads, el soporte de red y una API multiplataforma unificada para la gestión de ficheros.

3.3.6. BREW

BREW se usa para aplicaciones en dispositivos CDMA, aunque también soporta modelos GPRS/GSM. Las aplicaciones se distribuyen mediante una plataforma de contenido BREW y han tenido poca penetración en Europa. BREW puede proporcionar control completo del dispositivo y acceder a toda su funcionalidad. No obstante, el potencial que proporciona el código nativo con acceso directo a las APIs del dispositivo ha provocado que el proceso de desarrollo en BREW haya tenido que ser adaptado, en gran medida, para todos los vendedores de software reconocidos. Para que se pueda descargar en teléfonos normales, el software tiene que ser comprobado, probado y recibir la aprobación de Qualcomm mediante su programa de testeo TRUE BREW.

3.3.7. Palm OS

La plataforma Palm OS ha proporcionado a sus dispositivos móviles herramientas de negocio esenciales, así como la capacidad de acceder a Internet o a una base de datos central corporativa mediante una conexión inalámbrica. Palm webOS es el sistema operativo móvil propietario. Funciona sobre un kernel Linux que soporta multitarea.

3.4. Lenguajes de desarrollo de aplicaciones móviles

Un lenguaje de programación [13] es un lenguaje informático especialmente diseñado para describir el conjunto de acciones consecutivas o instrucciones que un equipo informático debe ejecutar.

3.4.1. Desarrollo nativo Android

Kotlin es el lenguaje de desarrollo para aplicaciones móviles que ayuda a prevenir errores de programación comunes en aplicaciones Android, siendo fuertemente tipado y permitiendo interoperar con código basado en Java.

Aun así, el lenguaje utilizado en el proyecto ha sido Java siendo el lenguaje de programación y la plataforma de desarrollo número uno. Java reduce los costes, acorta los plazos de desarrollo, impulsa la innovación y mejora los servicios de las aplicaciones.

3.4.2. Desarrollo nativo en iOS

Para iOS, Mac Apple TV y Apple Watch, Swift es el lenguaje utilizado por Apple para el desarrollo de sus aplicaciones. Es un lenguaje fuertemente tipado, que permite tener menos errores

durante el desarrollo, facilita el uso de patrones de programación funcional y también gestiona de forma automática la memoria.

Como se trata de un desarrollo nativo tiene un mayor rendimiento en dispositivos Apple, como ejemplo, en algoritmos de búsqueda tiene estimada una velocidad 2,6 veces mayor que Objective-C y hasta 8,4 veces mayor que Python.

3.4.3. Desarrollo Híbrido

Este entorno de desarrollo utiliza estándares web como HTML, CSS y JavaScript, permitiendo obtener una Progressive Web App (PWA) con la capacidad de trabajar en iOS y Android como una aplicación nativa.

No es necesario instalar una aplicación en el teléfono ya que permite acceder a través de un navegador, añadiendo la opción de generar aplicaciones para publicar tanto el Google Play como en la App Store.

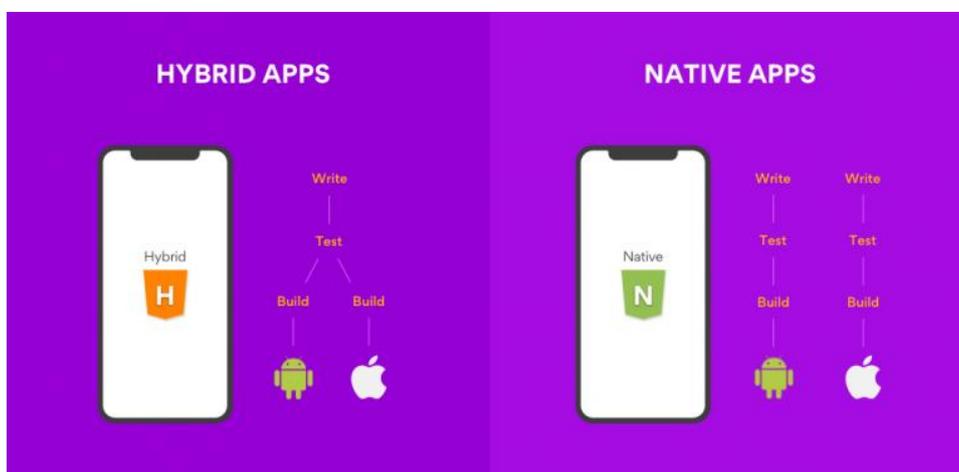


Ilustración 11. Comparativa apps híbridas y nativas

3.4.4. Apache Cordova

Apache Cordova [14] es un entorno de desarrollo libre diseñado para crear aplicaciones móviles utilizando CSS3, HTML5, y JavaScript en vez de utilizar APIs específicas de cada plataforma como Android, iOS, o Windows Phone.

Permite encapsular CSS, HTML, y código de Javascript dependiendo de la plataforma del dispositivo. Extiende las características de HTML y JavaScript para trabajar con el dispositivo. Las aplicaciones resultantes son híbridas, lo que significa que no son ni una aplicación móvil nativa (porque toda la representación gráfica se realiza vía vistas de Web en vez del *framework* nativo) ni puramente basadas en web (porque no son solo aplicaciones web, sino que están empaquetadas como aplicaciones para su distribución y tienen acceso a las APIs nativas del dispositivo). La mezcla de código nativo e híbrido ha sido posible desde la versión 1.9.

3.4.5. PhoneGap

PhoneGap [15] es un *framework* de código abierto creado para el desarrollo de aplicaciones móviles utilizando herramientas genéricas tales como JavaScript, HTML5 y CSS3 cuyas aplicaciones resultantes son híbridas.

PhoneGap maneja API que permiten tener acceso a elementos como el acelerómetro, la cámara, los contactos en el dispositivo, la red, el almacenamiento, las notificaciones, etc. Estas API se conectan al sistema operativo usando el código nativo del sistema huésped a través de una Interfaz de funciones foráneas en JavaScript.

3.4.6. Flutter

Flutter [16] es un SDK de código fuente abierto de desarrollo de aplicaciones móviles creado por Google. Suele usarse para desarrollar interfaces de usuario para aplicaciones en Android, iOS y Web así como método primario para crear aplicaciones para Google Fuchsia.

Al compilar para dispositivos móviles, Flutter usa el compilador Dart para convertir su código Dart en un código nativo que se ejecutará en la plataforma del dispositivo. Junto con un motor de renderizado personalizado para mostrar su interfaz de usuario dentro de una app móvil.

Las aplicaciones móviles de Flutter acceden a las funciones nativas del dispositivo mediante una biblioteca de complementos similar a Ionic y React Native. Flutter no usa los elementos de la interfaz de usuario nativos. Como encontrarías en React Native, ni utiliza componentes web como Ionic. En su lugar, Flutter ofrece su propia biblioteca de widgets de interfaz de usuario.

3.4.7. Ionic

Ionic [17] es un framework originalmente basado en Angular. Permite a los programadores el desarrollo de apps con tecnologías web. Utiliza estándares como HTML, CSS y JavaScript. Prepara el código de una app para que pueda funcionar tanto en plataformas iOS como en Android. También, ofrece la posibilidad de realizar una compilación más y hacer una ampliación de escritorio basada en ElectronJS. Dando lugar a una Progressive Web App.

Cuando se ejecuta en dispositivos móviles, Ionic se ejecuta dentro de un contenedor nativo utilizando Cordova o, más recientemente, Capacitor. Permite el acceso completo a cualquier API o características del dispositivo nativo.

Ionic utiliza el estándar de componentes web. Por lo que se ejecuta en cualquier navegador web y son compatibles con cualquier framework JS, incluidos React, Vue y Angular. Ionic proporciona una biblioteca de más de 100 componentes de interfaz de usuario que puede personalizar con CSS para adaptarse a las pautas de la marca. También puede usar Stencil.

3.4.8. Comparativa entre aplicaciones nativas e híbridas

A continuación se muestra la tabla comparativa entre aplicaciones nativas e híbridas.

	Aplicaciones híbridas	Aplicaciones nativas
Coste de desarrollo y mantenimiento	Coste menor al ser un único desarrollo para múltiples plataformas	Coste elevado debido a la programación por tipo de dispositivo con lenguajes más complejos
Rendimiento	Bastante bueno, excepto en aplicaciones que exigen de mucho hardware, como juegos	Óptimo
Acceso a características del dispositivo	Prácticamente completo	Completo
Tiempo de desarrollo	Mucho menor debido a que es un lenguaje mucho más simple y un único desarrollo por plataforma	Duplica como mínimo el tiempo a la híbrida debido a que serán dos desarrollos por separado

Tabla 5. Tabla comparativa aplicaciones híbridas y nativas

3.4.1. Comparativa lenguajes de programación

Para elegir el lenguaje de programación con el que desarrollar la aplicación, se ha realizado una tabla comparativa entre los diferentes lenguajes existentes. La mayoría de ellos tienen características similares, por tanto, se ha seleccionado Java como lenguaje usado en Android, ya que es un lenguaje de programación muy usado a lo largo de mi carrera y con el que es sencillo programar.

Lenguaje	Java	Kotlin	Swift	HTML, CSS, Javascript (híbrida)
Coste de desarrollo	Alto	Alto	Alto	Medio
Interfaz de usuario	Buena	Buena	Buena	Buena
Rendimiento	Bueno	Bueno	Bueno	Medio
Multiplataforma	Sí	Sí	Sí	Sí
Tiempo de desarrollo	Alto	Alto	Alto	Medio
App Stores	Sí	Sí	Sí	Sí

Tabla 6. Comparativa de lenguajes de programación

3.5. Bases de datos

Una base de datos [18] es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

3.5.1. MySQL

Se trata de un sistema de gestión de bases de datos relacional [19] desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo y una de las más populares en general junto a Oracle y Microsoft SQL Server, todo para entornos de desarrollo web. Está desarrollado en su mayor parte en ANSI C y C++. Tradicionalmente se considera uno de los cuatro componentes de la pila de desarrollo LAMP y WAMP.

3.5.2. PostgreSQL

PostgreSQL [20], también llamado Postgres, es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto, publicado bajo la licencia PostgreSQL, similar a la BSD o la MIT. Algunas de las principales características son la alta concurrencia y la amplia variedad de tipos nativos. También nos permite almacenar Json en nuestra base de datos, super útil cuando no es posible normalizar algunos datos. Lo que permite obtener todos los beneficios de las bases de datos relacionales, con un toque de flexibilidad.

Es multiplataforma, podemos utilizarlo tanto en Windows como GNU/Linux, MacOS, BSD y otras. Además es compatible con varios lenguajes de programación como C, C++, Java, Perl, PHP, Python, Ruby y más.

Otra de sus ventajas es que sigue los principios ACID de Atomicidad, Consistencia, Aislamiento (Isolation) y Durabilidad. Estos principios son fundamentales para la asegurar la fiabilidad de la información.

Cabe destacar que es Software libre, por lo que podemos utilizarlo sin ningún tipo de limitante. Está no pertenece a una empresa como sí pasa con MySQL. Los encargados de actualizar y mantener Postgres son la comunidad PGDG (PostgreSQL Global Development Group).

Ha sido la base de datos utilizada en este proyecto ya que su conexión con la aplicación es muy sencilla.

3.5.3. MongoDB

MongoDB [21] es la base de datos no-relacional más utilizada siendo de código abierto encontrándose bajo la licencia GNU AGPL v3.0. Fue desarrollada por MongoDB Inc y sigue siendo mantenida por esta misma compañía.

Al igual que Postgres, es compatible con varios lenguajes como JavaScript, Python, Java, PHP, Go, C++, C, Ruby y Perl. Así mismo, almacena la información en BSON, un formato binario de Json. BSON no es un formato que se pueda leer como Json, sino que luego es traducido a Json para su lectura.

Gracias a que es basado en documentos podemos modificar la estructura de la base de datos sin problema. Cada documento es una instancia diferente, por lo que se puede modificar su estructura, claves y demás. Esto es posible dado que las claves no están fuertemente relacionadas, sino que son una referencia.

3.5.4. MariaDB

MariaDB [22] es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL (General Public License). Este sistema introduce dos motores de almacenamiento nuevos, uno llamado Aria que reemplaza a MyISAM, y otro llamado XtraDB, en sustitución de InnoDB. Tiene una alta compatibilidad con MySQL ya que posee las mismas órdenes, interfaces, API y bibliotecas, siendo su objetivo poder cambiar un servidor por otro directamente.

Este SGBD surge a raíz de la compra de Sun Microsystems, compañía que había comprado previamente MySQL AB por parte de Oracle. MariaDB es una bifurcación directa de MySQL que asegura la existencia de una versión de este producto con licencia GPL. Widenius decidió crear esta variante porque estaba convencido de que el único interés de Oracle en MySQL era reducir la competencia que MySQL suponía para Oracle.

3.5.1. Comparativa de bases de datos

SQL son las bases de datos relacionales, estas manejan una estructura de datos ordenada. En este caso todos los grupos de datos tienen que ser iguales. Por ejemplo, que todos los objetos de tipo usuario tenga un nombre, apellido, fecha de nacimiento y estén relacionados con una o más intereses. Por supuesto podemos dejar campos vacíos, aunque no es lo ideal.

Las Bases de Datos SQL tienen muchas ventajas, si bien son más rígidas y no tan fáciles de escalar. Estas son más consistentes y robustas, su gran trayectoria nos da seguridad en la permanencia de los datos.

Gracias a su antigüedad también tenemos más herramientas para trabajar con este tipo de datos. Por consecuencia, más información al respecto y más personas especializadas en la materia.

Como una ventaja y a su vez desventaja tenemos la propiedad de la atomicidad, está lo que nos dice es que si la operación a la base de datos no se completa entonces se realiza un rollback llevando a consumir más recursos en el servidor.

NoSQL (Not Only SQL) utiliza un modelo no relacional, donde los esquemas son flexibles. Esto hace que sean más fáciles de diseñar, ya que no es necesario desarrollar un modelo con precisión como en el caso de las Bases de Datos relacionales.

Como ya sabemos los datos se almacenan en documentos con formato similar a Json. En los cuales no se tiene un identificador que relacione los datos, algo muy útil cuando no tenemos un esquema exacto.

Otra ventaja de este modelo de datos es que nos permite escalar horizontalmente, permitiendo almacenar la base de datos en un clúster. Incluso hay varios proveedores que ofrecen este servicio completamente administrado.

Con las Base de Datos NoSQL podemos obtener un alto rendimiento, especialmente para los modelos de datos para los cuales fueron diseñadas. Por supuesto esta diferencia se nota más cuando trabajamos a gran escala, en proyectos pequeños no notaremos la diferencia de performance respecto a las bases de datos relacionales.

	SQL	NoSQL
Carga de trabajo	Son óptimas para el procesamiento de transacciones online, cuando se necesita que las aplicaciones sean coherentes y transaccionales. Y para el procesamiento analítico online.	Permiten el uso de varios patrones de acceso a los datos, siendo ideales para aplicaciones de baja latencia. Están pensadas para el análisis sobre datos semiestructurados.
Modelo de datos	Las bases de datos SQL normalizan los datos en tablas con filas y columnas, definiendo los índices y las relaciones entre tablas. Ideal para modelos rígidos.	Existen varios modelos de datos NoSQL, como Documentos, Clave-valor, Gráficos, etc. Los cuales están optimizados para ser escalables y ofrecer el mejor rendimiento.
Rendimiento	Para lograr un buen rendimiento es necesario optimizar consultas, estructura de tablas e índices. Es más eficiente en la escritura.	Por lo general ofrecen un óptimo rendimiento para la lectura de datos. No es tan eficiente en la escritura.
Escalado	Se escalan de forma vertical, aumentando el hardware en un servidor. También se pueden realizar replicas para la lectura de datos.	Los datos se pueden particionar y dividir geográficamente. De igual manera, el escalado horizontal es super sencillo de realizar.

	SQL	NoSQL
API	La comunicación es mediante consultas en lenguaje SQL. Este se encuentra estandarizado hace varios años y es el mismo lenguaje para todas las bases de datos relacionales.	Existen APIs basadas en objetos para la interacción con la base de datos. Aun no existen estándares en este tipo de bases de datos.

Tabla 7. Comparativa entre bases de datos relacionales y no relacionales

Al haber elegido utilizar una base de datos relacional, se ha tenido que decidir entre usar MySQL o PostgreSQL. Acto seguido, se presenta una tabla con la comparativa de los servicios de base de datos MySQL y PostgreSQL.

	MySQL	PostgreSQL
Arquitectura	Cliente/Servidor	Cliente/Servidor
Sistemas operativos	FreeBSD, Linux, OS X, Solaris, Windows	FreeBSD, HP-UX, Linux, NetBSD, OS X, Solaris, Unix, Windows
Escalabilidad	Flexible y altamente escalable	Escalable y ajustable a las características del sistema
Consistencia transaccional	ACID	ACID
Soporte JSON	Sí	Sí
Casos de uso populares	Sitios web, aplicaciones web, LAMP stack	Minería de datos, analíticas, BI, Data Warehousing

Tabla 8. Comparativa entre MySQL y PostgreSQL

Se ha elegido PostgreSQL porque se trata de una base de datos utilizable en varios sistemas operativos, con alta escalabilidad y ajustable a las características del sistema. Es una base de datos muy fácil de usar, y su conexión con la aplicación Android es muy sencilla.

3.6. Servidores Web

3.6.1. Apache Tomcat

Apache Tomcat [23] funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Adicionalmente, implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Oracle Corporation (aunque creado por Sun Microsystems).

Como principales características, Tomcat soporta la especificación Servlet 3.0, que permite [24]:

- Soporte asíncrono.
- Configuración dinámica: fragmentos web (librerías pueden embeber partes de un web.xml de modo que no sea necesario añadirlos al web.xml de la aplicación.
- Soporta anotaciones: los filtros, Servlets y Listeners pueden definirse por anotaciones, sin necesidad de crear un web.xml.
- API Servlet extendida: permite añadir Servlets y Filter después del arranque de la aplicación.
- Modo embebido simplificado: Tomcat 7 incluye un API para esto (al estilo Jetty).
- Mejoras en Logging: con un formateador que escribe logs en una única línea.
- Alias: permite incluir ficheros externos dentro de una aplicación, como directorios de imágenes o Javascript, de modo que puedan ser compartidos entre todas las aplicaciones.
- Detección y prevención de Memory Leaks
- Mejoras en Seguridad

3.6.2. Nginx

Nginx [25] es un famoso software de servidor web de código abierto. En su versión inicial, funcionaba en servidores web HTTP. Sin embargo, hoy en día también sirve como proxy inverso, balanceador de carga HTTP y proxy de correo electrónico para IMAP, POP3 y SMTP.

A continuación, se muestra una breve comparativa entre Apache y Nginx:

- **Compatibilidad del sistema operativo:** Tanto NGINX como Apache pueden ejecutarse en muchos sistemas operativos que soportan el sistema Unix. Desafortunadamente, el rendimiento de NGINX en Windows no es tan bueno como en otras plataformas.

- **Soporte al usuario:** Aunque NGINX y Apache tienen soporte por correo y un foro de Stack Overflow, Apache carece de soporte por parte de su compañía.
- **Rendimiento:** NGINX puede ejecutar simultáneamente 1000 conexiones de contenido estático dos veces más rápido que Apache y usa un poco menos de memoria. Sin embargo, cuando se comparan por su rendimiento en la ejecución de contenido dinámico, ambos tienen la misma velocidad.

3.6.3. LiteSpeed

Es un servidor [26] que incorpora muchas de las cosas buenas de Apache, es compatible con una buena parte de la configuración y algunos módulos importantes de Apache que lo convierten en un sustituto ideal.

La diferencia fundamental es la velocidad y la eficiencia, tiene una arquitectura diseñada para aprovechar al máximo el uso de CPU y la cantidad de núcleos de este, eso sin descuidar el uso de memoria, ambas cosas implican por un lado una menor carga de trabajo para el servidor y el poder manejar un tráfico mayor con el mismo hardware que lo que podríamos hacer con Apache.

Una de las grandes ventajas que lo transforman en el sustituto ideal de Apache es su compatibilidad con los archivos .htaccess de Apache que garantiza una migración automática de Apache a LiteSpeed sin necesidad de ajustes engorrosos.

3.6.4. XAMPP

XAMPP [27] es una aplicación de gestión de servidores que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X. XAMPP solamente requiere descargar y ejecutar un archivo .zip, .tar, o .exe, con unas pequeñas configuraciones en alguno de sus componentes que el servidor Web necesitará.

XAMPP ha sido usado en este proyecto para desplegar el servidor Java en él y poder realizar conexiones con la base de datos y con la aplicación Android de una manera más sencilla.

3.6.5. Insomnia

Insomnia [28] es un cliente GraphQL y REST de código abierto destinado a facilitar las pruebas y la depuración de las API. Insomnia gestiona las solicitudes de una manera simple y elegante, proporcionando las siguientes características básicas:

- Constructor de parámetros de cadenas de consulta.

- Organizar las solicitudes en espacios de trabajo y grupos.
- Ejecutar las solicitudes con un solo clic.
- Fragmentos reutilizables.
- Plantillas Swig.
- Validación de JSON.
- Resaltado de sintaxis (JSON, javascript, XML, CSS, ...).
- Ayudante de autenticación básica HTTP.
- Importación/exportación.
- Respuesta en cualquier tamaño de ventana.
- Cambio rápido de peticiones (estilo ctrl-p).
- Codificación y decodificación de url.
- Codificación y decodificación base64.
- Exportación como curl.

Esta herramienta ha sido utilizada para gestionar las solicitudes del servidor de una manera más sencilla y elegante, teniendo todas las solicitudes organizadas en dicha herramienta.

4. Desarrollo

En este capítulo, se va a desarrollar la implementación y el diseño del proyecto, explicando la solución propuesta, el desarrollo del sistema con sus conexiones, así como todo el funcionamiento y diseño de la aplicación Android.

4.1. Solución propuesta

La solución propuesta es una aplicación Android intuitiva, fácil de usar y capaz de automatizar todos los procesos arbitrales necesarios para reducir los errores humanos.

Los datos para tener en cuenta son la información de los árbitros, la información de los jugadores de los equipos, la información de los encuentros teniendo en cuenta la fecha, hora, lugar, cancha, división, etc. Adicionalmente, se tiene en cuenta los resultados obtenidos durante el partido, llevando a cabo el acta en tiempo real.

La aplicación dispone de un apartado para seleccionar el encuentro junto con los jugadores que jugarán el partido. A su vez, existe un apartado para realizar el partido en tiempo real, enseñando las faltas de rotación de los diferentes sistemas. Por último, hay un apartado que muestra el acta rellena en tiempo real y procede a descargarla en su teléfono cada vez que se accede a dicho apartado.

En conclusión, la aplicación Android manipulará una serie de datos de los encuentros, digitalizando los procesos y facilitando el trabajo a los árbitros.

4.2. Arquitectura del proyecto

En este apartado, se describe la arquitectura del sistema, así como las relaciones entre los diferentes elementos que lo componen. A continuación, se muestra el diagrama de los componentes del sistema:

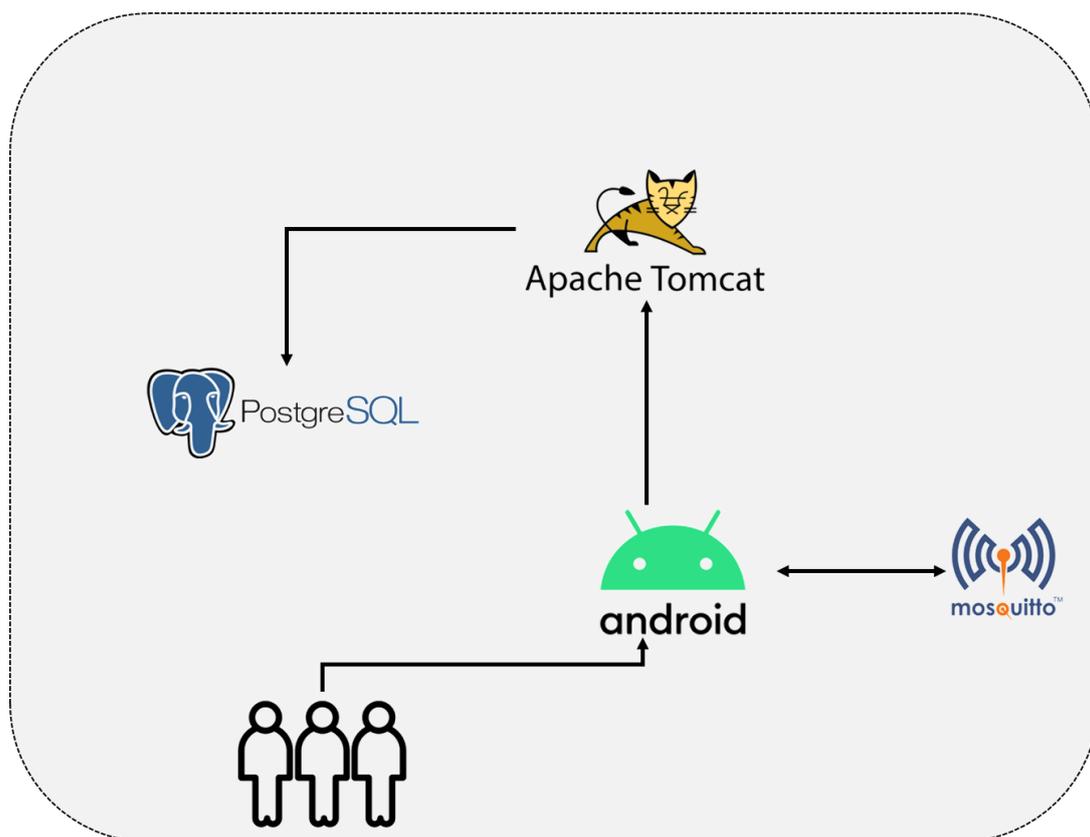


Ilustración 12. Diagrama de elementos

Como puede apreciarse en la imagen anterior, el sistema está compuesto por 4 elementos principales: una base de datos donde se almacenará la información, un contenedor Web donde ayudará a gestionar las peticiones de información, un bróker MQTT que ayudará a mantener actualizada la información en tiempo real y una aplicación Android. A continuación se detalla cada uno de estos elementos.

4.2.1. Base de datos en PostgreSQL

De cara a almacenar la información de los partidos y las actas realizadas de forma persistente, se ha creado una base de datos en PostgreSQL que almacena y gestiona la información necesaria para la ejecución de la aplicación. La base de datos ha sido creada con el nombre “playvolleyball” y consta de siete tablas que son: equipo, entrenador, jugador, partido, árbitros, acta y login. En la siguiente imagen puede verse el modelo de tablas desarrollado.

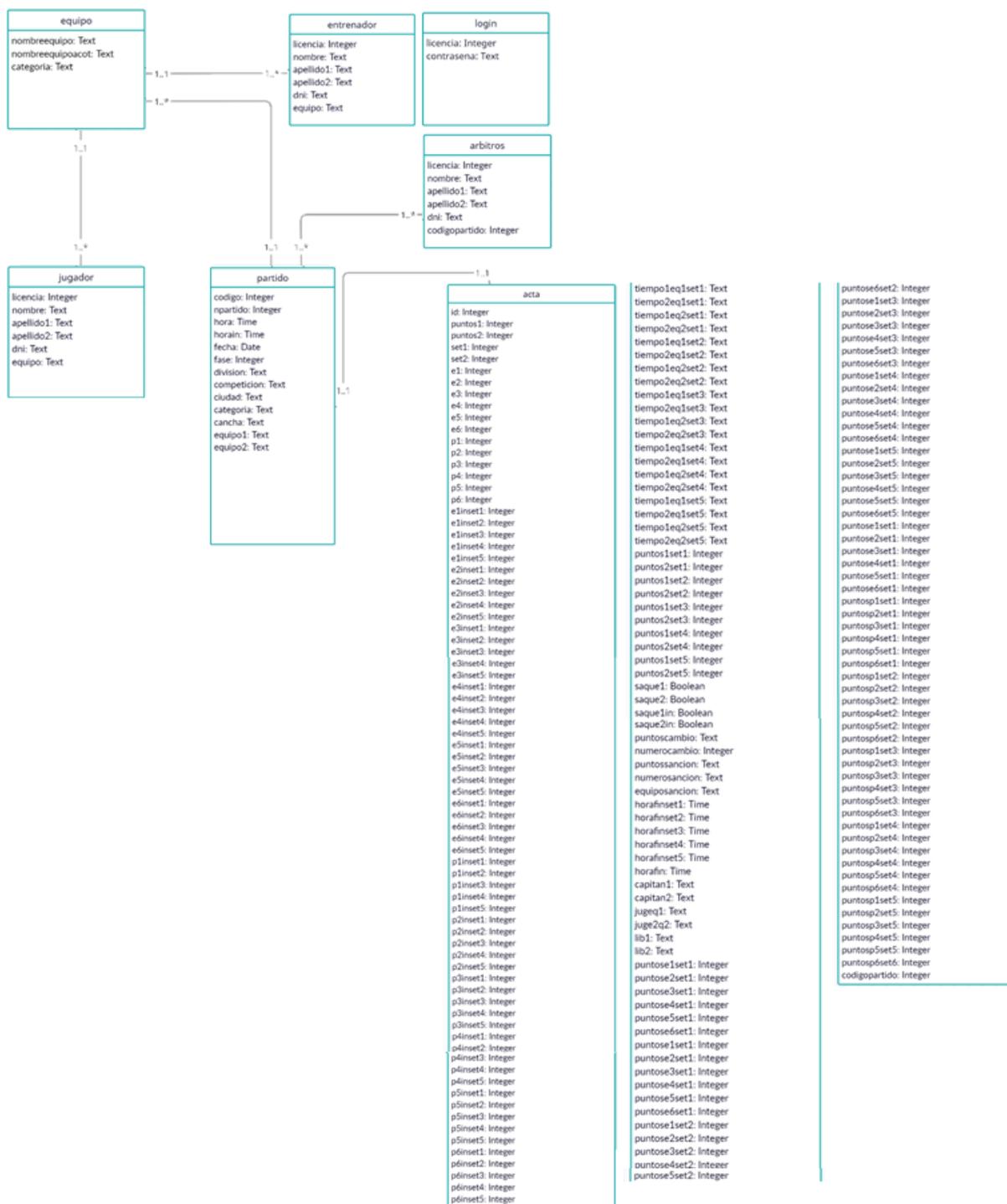


Ilustración 13. Diagrama de componentes

La tabla *jugador* almacena toda la información relacionada con los jugadores que forman parte de un equipo. Los datos que contiene la tabla son:

- **Licencia:** Licencia del jugador que se encuentra registrada en la Federación cuyo tipo es *Integer* y *not null*.
- **Nombre:** Nombre del jugador que se encuentra registrado en la Federación cuyo tipo es *Text* y *not null*.
- **Apellido1:** Primer apellido del jugador que se encuentra registrado en la Federación cuyo tipo es *Text* y *not null*.
- **Apellido2:** Segundo apellido del jugador que se encuentra registrado en la Federación cuyo tipo es *Text*.
- **Dni:** Documento de identidad que se encuentra registrado en la Federación cuyo tipo es *Text* y *not null*.
- **Numero:** Dorsal del jugador cuyo tipo es *Integer* y *not null*.
- **Equipo:** equipo del que forma parte el jugador siendo una clave foránea (FK) que hace referencia a la tabla equipo de manera que un jugador forma parte de un equipo determinado.

La tabla *entrenador* almacena toda la información relacionada con los entrenadores que forman parte de un equipo. Los datos que contiene la tabla son:

- **Licencia:** Licencia del entrenador que se encuentra registrada en la Federación cuyo tipo es *Integer* y *not null*.
- **Nombre:** Nombre del entrenador que se encuentra registrado en la Federación cuyo tipo es *Text* y *not null*.
- **Apellido1:** Primer apellido del entrenador que se encuentra registrado en la Federación cuyo tipo es *Text* y *not null*.
- **Apellido2:** Segundo apellido del entrenador que se encuentra registrado en la Federación cuyo tipo es *Text*.
- **Dni:** Documento de identidad que se encuentra registrado en la Federación cuyo tipo es *Text* y *not null*.
- **Equipo:** equipo del que forma parte el entrenador siendo una clave foránea (FK) que hace referencia a la tabla equipo de manera que un entrenador forma parte de un equipo determinado.

La tabla *equipo* almacena toda la información relacionada con los jugadores, entrenadores y la categoría a la que pertenecen. Los datos que contiene la tabla son:

- **Nombreequipo:** Nombre del equipo que se encuentra registrado en la Federación cuyo tipo es *Text* y *not null*.
- **Nombreequipoacot:** Nombre del equipo acotado cogiendo solo las tres primeras letras del nombre cuyo tipo es *Text* y *not null*.
- **Categoría:** Categoría en la que compite dicho equipo cuyo tipo es *Text* y *not null*.

La tabla *login* almacena toda la información para registrar y acceder a la información. Los datos que contiene la tabla son:

- **Licencia:** Licencia del entrenador o arbitro que se registre en la aplicación cuyo tipo es *Integer* y *not null*.
- **Contraseña:** Contraseña creada por el árbitro o el entrenador para acceder a la aplicación de forma segura cuyo tipo es *Text* y *not null*.

La tabla *arbitros* almacena toda la información relacionada con los árbitros que forman parte de un partido. Los datos que contiene la tabla son:

- **Licencia:** Licencia del árbitro que se encuentra registrada en la Federación cuyo tipo es *Integer* y *not null*.
- **Nombre:** Nombre del árbitro que se encuentra registrado en la Federación cuyo tipo es *Text* y *not null*.
- **Apellido1:** Primer apellido del árbitro que se encuentra registrado en la Federación de tipo *Text* y *not null*.
- **Apellido2:** Segundo apellido del árbitro que se encuentra registrado en la Federación cuyo tipo es *Text*.
- **Dni:** Documento de identidad que se encuentra registrado en la Federación cuyo tipo es *Text* y *not null*.
- **Codigopartido:** código del partido del que forma parte siendo una clave foránea (FK) que hace referencia a la tabla partido de manera que un árbitro puede formar parte de uno o varios partidos.

La tabla *partido* almacena toda la información relacionada con los partidos. Los datos que contiene la tabla son:

- **Código:** Código del partido que se encuentra registrado en la Federación cuyo tipo es *Integer* y *not null*.

- **Npartido:** Número de partido que se encuentra registrado en la Federación cuyo tipo es *Integer* y *not null*.
- **Hora:** Hora supuesta a la que debería empezar el partido registrada en la Federación cuyo tipo es *Time* y *not null*.
- **Horain:** Hora real de comienzo del partido cuyo tipo es *Date* y *not null*.
- **Fecha:** Fecha del encuentro que se encuentra registrada en la Federación cuyo tipo es *Date* y *not null*.
- **Division:** División en la que se disputa el partido encontrándose registrada en la Federación cuyo tipo es *Text* y *not null*.
- **Competicion:** Competición en la que se disputa el partido encontrándose registrado en la Federación cuyo tipo es *Text* y *not null*.
- **Ciudad:** Ciudad en la que se disputa el partido encontrándose registrada en la Federación cuyo tipo es *Text* y *not null*.
- **Categoria:** Categoría en la que se disputa el partido encontrándose registrada en la Federación cuyo tipo es *Text* y *not null*.
- **Cancha:** Cancha en la que se disputa el partido encontrándose registrada en la Federación cuyo tipo es *Text* y *not null*.
- **Equipo1:** primer equipo del partido del que forma parte siendo una clave foránea (FK) que hace referencia a la tabla equipo de manera que un partido está formado por este equipo.
- **Equipo2:** segundo equipo del partido del que forma parte siendo una clave foránea (FK) que hace referencia a la tabla equipo de manera que un partido está formado por este equipo.

La tabla *acta* almacena toda la información relacionada con los valores que se pueden modificar durante un partido. Estos valores se recogen en una tabla ya que son valores fijos y son obligatorios reflejarlos en el acta, por eso se ponen en una única tabla en lugar de dividirlo en varias tablas. Los datos que contiene la tabla son:

- **id:** Identificador del acta cuyo tipo es *Integer* y *not null*.
- **puntos1:** Puntos del marcador del equipo en el lado izquierdo del campo cuyo tipo es *Integer* y *not null*.
- **puntos2:** Puntos del marcador del equipo en el lado derecho del campo de tipo *Integer* y *not null*.

- **set1:** El número de sets conseguidos por parte del equipo en el lado izquierdo del cuyo tipo es *Integer* y *not null*.
- **set2:** El número de sets conseguidos por parte del equipo en el lado derecho del campo cuyo tipo es *Integer* y *not null*.
- **e1:** El número que se encuentra en la posición 1 del campo del equipo izquierdo cuyo tipo es *Integer* y *not null*.
- **e2:** El número que se encuentra en la posición 2 del campo del equipo izquierdo cuyo tipo es *Integer* y *not null*.
- **e3:** El número que se encuentra en la posición 3 del campo del equipo izquierdo cuyo tipo es *Integer* y *not null*.
- **e4:** El número que se encuentra en la posición 4 del campo del equipo izquierdo cuyo tipo es *Integer* y *not null*.
- **e5:** El número que se encuentra en la posición 5 del campo del equipo izquierdo cuyo tipo es *Integer* y *not null*.
- **e6:** El número que se encuentra en la posición 6 del campo del equipo izquierdo cuyo tipo es *Integer* y *not null*.
- **p1:** El número que se encuentra en la posición 1 del campo del equipo derecho cuyo tipo es *Integer* y *not null*.
- **p2:** El número que se encuentra en la posición 2 del campo del equipo derecho cuyo tipo es *Integer* y *not null*.
- **p3:** El número que se encuentra en la posición 3 del campo del equipo derecho cuyo tipo es *Integer* y *not null*.
- **p4:** El número que se encuentra en la posición 4 del campo del equipo derecho cuyo tipo es *Integer* y *not null*.
- **p5:** El número que se encuentra en la posición 5 del campo del equipo derecho cuyo tipo es *Integer* y *not null*.
- **p6:** El número que se encuentra en la posición 6 del campo del equipo derecho cuyo tipo es *Integer* y *not null*.
- **e1inset1:** El número inicial que se encuentra en la posición 1 del campo del equipo izquierdo en el primer set cuyo tipo es *Integer* y *not null*.
- **e1inset2:** El número inicial que se encuentra en la posición 1 del campo del equipo izquierdo en el segundo set cuyo tipo es *Integer* y *not null*.
- **e1inset3:** El número inicial que se encuentra en la posición 1 del campo del equipo izquierdo en el tercer set cuyo tipo es *Integer* y *not null*.

- **e1inset4:** El número inicial que se encuentra en la posición 1 del campo del equipo izquierdo en el cuarto set cuyo tipo es *Integer*.
- **e1inset5:** El número inicial que se encuentra en la posición 2 del campo del equipo izquierdo en el quinto set cuyo tipo es *Integer*.
- **e2inset1:** El número inicial que se encuentra en la posición 2 del campo del equipo izquierdo en el primer set cuyo tipo es *Integer* y *not null*.
- **e2inset2:** El número inicial que se encuentra en la posición 2 del campo del equipo izquierdo en el segundo set cuyo tipo es *Integer* y *not null*.
- **e2inset3:** El número inicial que se encuentra en la posición 2 del campo del equipo izquierdo en el tercer set cuyo tipo es *Integer* y *not null*.
- **e2inset4:** El número inicial que se encuentra en la posición 2 del campo del equipo izquierdo en el cuarto set cuyo tipo es *Integer*.
- **e2inset5:** El número inicial que se encuentra en la posición 2 del campo del equipo izquierdo en el quinto set cuyo tipo es *Integer*.
- **e3inset1:** El número inicial que se encuentra en la posición 3 del campo del equipo izquierdo en el primer set cuyo tipo es *Integer* y *not null*.
- **e3inset2:** El número inicial que se encuentra en la posición 3 del campo del equipo izquierdo en el segundo set cuyo tipo es *Integer* y *not null*.
- **e3inset3:** El número inicial que se encuentra en la posición 3 del campo del equipo izquierdo en el tercer set cuyo tipo es *Integer* y *not null*.
- **e3inset4:** El número inicial que se encuentra en la posición 3 del campo del equipo izquierdo en el cuarto set cuyo tipo es *Integer*.
- **e3inset5:** El número inicial que se encuentra en la posición 3 del campo del equipo izquierdo en el quinto set cuyo tipo es *Integer*.
- **e4inset1:** El número inicial que se encuentra en la posición 4 del campo del equipo izquierdo en el primer set cuyo tipo es *Integer* y *not null*.
- **e4inset2:** El número inicial que se encuentra en la posición 4 del campo del equipo izquierdo en el segundo set cuyo tipo es *Integer* y *not null*.
- **e4inset3:** El número inicial que se encuentra en la posición 4 del campo del equipo izquierdo en el tercer set cuyo tipo es *Integer* y *not null*.
- **e4inset4:** El número inicial que se encuentra en la posición 4 del campo del equipo izquierdo en el cuarto set cuyo tipo es *Integer*.
- **e4inset5:** El número inicial que se encuentra en la posición 4 del campo del equipo izquierdo en el quinto set cuyo tipo es *Integer*.

- **e5inset1:** El número inicial que se encuentra en la posición 5 del campo del equipo izquierdo en el primer set cuyo tipo es *Integer* y *not null*.
- **e5inset2:** El número inicial que se encuentra en la posición 5 del campo del equipo izquierdo en el segundo set cuyo tipo es *Integer* y *not null*.
- **e5inset3:** El número inicial que se encuentra en la posición 5 del campo del equipo izquierdo en el tercero set cuyo tipo es *Integer* y *not null*.
- **e5inset4:** El número inicial que se encuentra en la posición 5 del campo del equipo izquierdo en el cuarto set cuyo tipo es *Integer*.
- **e5inset5:** El número inicial que se encuentra en la posición 5 del campo del equipo izquierdo en el quinto set cuyo tipo es *Integer*.
- **e6inset1:** El número inicial que se encuentra en la posición 6 del campo del equipo izquierdo en el primer set cuyo tipo es *Integer* y *not null*.
- **e6inset2:** El número inicial que se encuentra en la posición 6 del campo del equipo izquierdo en el segundo set cuyo tipo es *Integer* y *not null*.
- **e6inset3:** El número inicial que se encuentra en la posición 6 del campo del equipo izquierdo en el tercero set cuyo tipo es *Integer* y *not null*.
- **e6inset4:** El número inicial que se encuentra en la posición 6 del campo del equipo izquierdo en el cuarto set cuyo tipo es *Integer*.
- **e6inset5:** El número inicial que se encuentra en la posición 6 del campo del equipo izquierdo en el quinto set cuyo tipo es *Integer*.
- **p1inset1:** El número inicial que se encuentra en la posición 1 del campo del equipo derecho en el primer set cuyo tipo es *Integer* y *not null*.
- **p1inset2:** El número inicial que se encuentra en la posición 1 del campo del equipo derecho en el segundo set cuyo tipo es *Integer* y *not null*.
- **p1inset3:** El número inicial que se encuentra en la posición 1 del campo del equipo derecho en el tercero set cuyo tipo es *Integer* y *not null*.
- **p1inset4:** El número inicial que se encuentra en la posición 1 del campo del equipo derecho en el cuarto set cuyo tipo es *Integer*.
- **p1inset5:** El número inicial que se encuentra en la posición 2 del campo del equipo derecho en el quinto set cuyo tipo es *Integer*.
- **p2inset1:** El número inicial que se encuentra en la posición 2 del campo del equipo derecho en el primer set cuyo tipo es *Integer* y *not null*.
- **p2inset2:** El número inicial que se encuentra en la posición 2 del campo del equipo derecho en el segundo set cuyo tipo es *Integer* y *not null*.

- **p2inset3:** El número inicial que se encuentra en la posición 2 del campo del equipo derecho en el tercero set cuyo tipo es *Integer* y *not null*.
- **p2inset4:** El número inicial que se encuentra en la posición 2 del campo del equipo derecho en el cuarto set cuyo tipo es *Integer*.
- **p2inset5:** El número inicial que se encuentra en la posición 2 del campo del equipo derecho en el quinto set cuyo tipo es *Integer*.
- **p3inset1:** El número inicial que se encuentra en la posición 3 del campo del equipo derecho en el primer set cuyo tipo es *Integer* y *not null*.
- **p3inset2:** El número inicial que se encuentra en la posición 3 del campo del equipo derecho en el segundo set cuyo tipo es *Integer* y *not null*.
- **p3inset3:** El número inicial que se encuentra en la posición 3 del campo del equipo derecho en el tercero set cuyo tipo es *Integer* y *not null*.
- **p3inset4:** El número inicial que se encuentra en la posición 3 del campo del equipo derecho en el cuarto set cuyo tipo es *Integer*.
- **p3inset5:** El número inicial que se encuentra en la posición 3 del campo del equipo derecho en el quinto set cuyo tipo es *Integer*.
- **p4inset1:** El número inicial que se encuentra en la posición 4 del campo del equipo derecho en el primer set cuyo tipo es *Integer* y *not null*.
- **p4inset2:** El número inicial que se encuentra en la posición 4 del campo del equipo derecho en el segundo set cuyo tipo es *Integer* y *not null*.
- **p4inset3:** El número inicial que se encuentra en la posición 4 del campo del equipo derecho en el tercero set cuyo tipo es *Integer* y *not null*.
- **p4inset4:** El número inicial que se encuentra en la posición 4 del campo del equipo derecho en el cuarto set cuyo tipo es *Integer*.
- **p4inset5:** El número inicial que se encuentra en la posición 4 del campo del equipo derecho en el quinto set cuyo tipo es *Integer*.
- **p5inset1:** El número inicial que se encuentra en la posición 5 del campo del equipo derecho en el primer set cuyo tipo es *Integer* y *not null*.
- **p5inset2:** El número inicial que se encuentra en la posición 5 del campo del equipo derecho en el segundo set cuyo tipo es *Integer* y *not null*.
- **p5inset3:** El número inicial que se encuentra en la posición 5 del campo del equipo derecho en el tercero set cuyo tipo es *Integer* y *not null*.
- **p5inset4:** El número inicial que se encuentra en la posición 5 del campo del equipo derecho en el cuarto set cuyo tipo es *Integer*.

- **p5inset5**: El número inicial que se encuentra en la posición 5 del campo del equipo derecho en el quinto set cuyo tipo es *Integer*.
- **p6inset1**: El número inicial que se encuentra en la posición 6 del campo del equipo derecho en el primer set cuyo tipo es *Integer* y *not null*.
- **p6inset2**: El número inicial que se encuentra en la posición 6 del campo del equipo derecho en el segundo set cuyo tipo es *Integer* y *not null*.
- **p6inset3**: El número inicial que se encuentra en la posición 6 del campo del equipo derecho en el tercero set cuyo tipo es *Integer* y *not null*.
- **p6inset4**: El número inicial que se encuentra en la posición 6 del campo del equipo derecho en el cuarto set cuyo tipo es *Integer*.
- **p6inset5**: El número inicial que se encuentra en la posición 6 del campo del equipo derecho en el quinto set cuyo tipo es *Integer*.
- **tiempo1eq1set1**: Primer tiempo solicitado por el equipo izquierdo en el primer set cuyo tipo es *Text* ya que obtiene un *String* con datos.
- **tiempo2eq1set1**: Segundo tiempo solicitado por el equipo izquierdo en el primer set cuyo tipo es *Text* ya que obtiene un *String* con datos.
- **tiempo1eq2set1**: Primer tiempo solicitado por el equipo derecho en el primer set cuyo tipo es *Text* ya que obtiene un *String* con datos.
- **tiempo2eq2set1**: Segundo tiempo solicitado por el equipo derecho en el primer set cuyo tipo es *Text* ya que obtiene un *String* con datos.
- **tiempo1eq1set2**: Primer tiempo solicitado por el equipo izquierdo en el segundo set cuyo tipo es *Text* ya que obtiene un *String* con datos.
- **tiempo2eq1set2**: Segundo tiempo solicitado por el equipo izquierdo en el segundo set cuyo tipo es *Text* ya que obtiene un *String* con datos.
- **tiempo1eq2set2**: Primer tiempo solicitado por el equipo derecho en el segundo set cuyo tipo es *Text* ya que obtiene un *String* con datos.
- **tiempo2eq2set2**: Segundo tiempo solicitado por el equipo derecho en el segundo set cuyo tipo es *Text* ya que obtiene un *String* con datos.
- **tiempo1eq1set3**: Primer tiempo solicitado por el equipo izquierdo en el tercer set cuyo tipo es *Text* ya que obtiene un *String* con datos.
- **tiempo2eq1set3**: Segundo tiempo solicitado por el equipo izquierdo en el tercer set cuyo tipo es *Text* ya que obtiene un *String* con datos.
- **tiempo1eq2set3**: Primer tiempo solicitado por el equipo derecho en el tercer set cuyo tipo es *Text* ya que obtiene un *String* con datos.

- **tiempo2eq2set3:** Segundo tiempo solicitado por el equipo derecho en el tercero set cuyo tipo es *Text* ya que obtiene un String con datos.
- **tiempo1eq1set4:** Primer tiempo solicitado por el equipo izquierdo en el cuarto set cuyo tipo es *Text* ya que obtiene un String con datos.
- **tiempo2eq1set4:** Segundo tiempo solicitado por el equipo izquierdo en el cuarto set cuyo tipo es *Text* ya que obtiene un String con datos.
- **tiempo1eq2set4:** Primer tiempo solicitado por el equipo derecho en el cuarto set cuyo tipo es *Text* ya que obtiene un String con datos.
- **tiempo2eq2set4:** Segundo tiempo solicitado por el equipo derecho en el cuarto set cuyo tipo es *Text* ya que obtiene un String con datos.
- **tiempo1eq1set5:** Primer tiempo solicitado por el equipo izquierdo en el quinto set cuyo tipo es *Text* ya que obtiene un String con datos.
- **tiempo2eq1set5:** Segundo tiempo solicitado por el equipo izquierdo en el quinto set cuyo tipo es *Text* ya que obtiene un String con datos.
- **tiempo1eq2set5:** Primer tiempo solicitado por el equipo derecho en el quinto set cuyo tipo es *Text* ya que obtiene un String con datos.
- **tiempo2eq2set5:** Segundo tiempo solicitado por el equipo derecho en el quinto set cuyo tipo es *Text* ya que obtiene un String con datos.
- **puntos1set1:** Puntos finales del equipo izquierdo en el primer set cuyo tipo es *Integer* y *not null*.
- **puntos2set1:** Puntos finales del equipo derecho en el primer set cuyo tipo es *Integer* y *not null*.
- **puntos1set2:** Puntos finales del equipo izquierdo en el segundo set cuyo tipo es *Integer* y *not null*.
- **puntos2set2:** Puntos finales del equipo derecho en el segundo set cuyo tipo es *Integer* y *not null*.
- **puntos1set3:** Puntos finales del equipo izquierdo en el tercer set cuyo tipo es *Integer* y *not null*.
- **puntos2set3:** Puntos finales del equipo derecho en el tercer set cuyo tipo es *Integer* y *not null*.
- **puntos1set4:** Puntos finales del equipo izquierdo en el cuarto set cuyo tipo es *Integer* y *not null*.
- **puntos2set4:** Puntos finales del equipo derecho en el cuarto set cuyo tipo es *Integer* y *not null*.

- **puntos1set5:** Puntos finales del equipo izquierdo en el quinto set cuyo tipo es *Integer* y *not null*.
- **puntos2set5:** Puntos finales del equipo derecho en el quinto set cuyo tipo es *Integer* y *not null*.
- **saque1:** Saque actual del equipo izquierdo cuyo tipo es *Boolean* y *not null*.
- **saque2:** Saque actual del equipo derecho cuyo tipo es *Boolean* y *not null*.
- **saque1in:** Indica si ha ganado el equipo izquierdo el sorteo de saque cuyo tipo es *Boolean* y *not null*.
- **saque2in:** Indica si ha ganado el equipo izquierdo el sorteo de saque cuyo tipo es *Boolean* y *not null*.
- **puntoscambio:** Indica el marcador cuando se ha realizado un cambio cuyo tipo es *Text* ya que se obtiene un String con varios datos.
- **numerocambio:** Indica el número por el que se sustituye cuyo tipo es *Text*.
- **puntossancion:** Indica el marcador cuando se ha realizado una sanción cuyo tipo es *Text* ya que se obtiene un String con varios datos.
- **numerosancion:** Indica a quién va dirigida la sanción, siendo este “C” de entrenador o un numero en caso de que sea un jugador y cuyo tipo es *Text*.
- **equiposancion:** Indica el equipo al que va dirigida la sanción siendo este “A”² o “B”³ y cuyo tipo es *Text*.
- **horafinset1:** Indica la hora a la que ha finalizado el primer set y cuyo tipo es *Time* y *not null*.
- **horafinset2:** Indica la hora a la que ha finalizado el primer set y cuyo tipo es *Time* y *not null*.
- **horafinset3:** Indica la hora a la que ha finalizado el primer set y cuyo tipo es *Time* y *not null*.
- **horafinset4:** Indica la hora a la que ha finalizado el primer set y cuyo tipo es *Time*.
- **horafinset5:** Indica la hora a la que ha finalizado el primer set y cuyo tipo es *Time*.
- **horafin:** Indica la hora a la que ha finalizado el partido set y cuyo tipo es *Time* y *not null*.

² Equipo del lado izquierdo del campo

³ Equipo del lado derecho del campo

- **capitan1**: Indica el primer apellido del capitán del equipo local y cuyo tipo es *Text* y *not null*.
- **capitan2**: Indica el primer apellido del capitán del equipo visitante y cuyo tipo es *Text* y *not null*.
- **jugeq1**: Indica todos los apellidos del equipo local que van a jugar el encuentro y cuyo tipo es *Text* y *not null*.
- **jugeq2**: Indica todos los apellidos del equipo visitante que van a jugar el encuentro y cuyo tipo es *Text* y *not null*.
- **lib1**: Indica todos los apellidos del equipo local que van a jugar el encuentro con rol de líbero y cuyo tipo es *Text* y *not null*.
- **lib2**: Indica todos los apellidos del equipo visitante que van a jugar el encuentro con rol de líbero y cuyo tipo es *Text* y *not null*.
- **puntose1set1**: Puntos conseguidos por el jugador que comienza en la posición 1 en el primer set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose2set1**: Puntos conseguidos por el jugador que comienza en la posición 2 en el primer set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose3set1**: Puntos conseguidos por el jugador que comienza en la posición 3 en el primer set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose4set1**: Puntos conseguidos por el jugador que comienza en la posición 4 en el primer set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose5set1**: Puntos conseguidos por el jugador que comienza en la posición 5 en el primer set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose6set1**: Puntos conseguidos por el jugador que comienza en la posición 6 en el primer set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose1set2**: Puntos conseguidos por el jugador que comienza en la posición 1 en el segundo set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose2set2**: Puntos conseguidos por el jugador que comienza en la posición 2 en el segundo set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose3set2**: Puntos conseguidos por el jugador que comienza en la posición 3 en el segundo set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.

- **puntose4set2:** Puntos conseguidos por el jugador que comienza en la posición 4 en el segundo set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose5set2:** Puntos conseguidos por el jugador que comienza en la posición 5 en el segundo set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose6set2:** Puntos conseguidos por el jugador que comienza en la posición 6 en el segundo set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose1set3:** Puntos conseguidos por el jugador que comienza en la posición 1 en el tercer set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose2set3:** Puntos conseguidos por el jugador que comienza en la posición 2 en el tercer set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose3set3:** Puntos conseguidos por el jugador que comienza en la posición 3 en el tercer set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose4set3:** Puntos conseguidos por el jugador que comienza en la posición 4 en el tercer set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose5set3:** Puntos conseguidos por el jugador que comienza en la posición 5 en el tercer set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose6set3:** Puntos conseguidos por el jugador que comienza en la posición 6 en el tercer set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose1set4:** Puntos conseguidos por el jugador que comienza en la posición 1 en el cuarto set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose2set4:** Puntos conseguidos por el jugador que comienza en la posición 2 en el cuarto set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose3set4:** Puntos conseguidos por el jugador que comienza en la posición 3 en el cuarto set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose4set4:** Puntos conseguidos por el jugador que comienza en la posición 4 en el cuarto set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose5set4:** Puntos conseguidos por el jugador que comienza en la posición 5 en el cuarto set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose6set4:** Puntos conseguidos por el jugador que comienza en la posición 6 en el cuarto set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.

- **puntose1set5**: Puntos conseguidos por el jugador que comienza en la posición 1 en el quinto set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose2set5**: Puntos conseguidos por el jugador que comienza en la posición 2 en el quinto set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose3set5**: Puntos conseguidos por el jugador que comienza en la posición 3 en el quinto set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose4set5**: Puntos conseguidos por el jugador que comienza en la posición 4 en el quinto set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose5set5**: Puntos conseguidos por el jugador que comienza en la posición 5 en el quinto set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntose6set5**: Puntos conseguidos por el jugador que comienza en la posición 6 en el quinto set del equipo posicionado en la izquierda y cuyo tipo es *Integer*.
- **puntosp1set1**: Puntos conseguidos por el jugador que comienza en la posición 1 en el primer set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp2set1**: Puntos conseguidos por el jugador que comienza en la posición 2 en el primer set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp3set1**: Puntos conseguidos por el jugador que comienza en la posición 3 en el primer set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp4set1**: Puntos conseguidos por el jugador que comienza en la posición 4 en el primer set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp5set1**: Puntos conseguidos por el jugador que comienza en la posición 5 en el primer set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp6set1**: Puntos conseguidos por el jugador que comienza en la posición 6 en el primer set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp1set2**: Puntos conseguidos por el jugador que comienza en la posición 1 en el segundo set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp2set2**: Puntos conseguidos por el jugador que comienza en la posición 2 en el segundo set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp3set2**: Puntos conseguidos por el jugador que comienza en la posición 3 en el segundo set del equipo posicionado en la derecha y cuyo tipo es *Integer*.

- **puntosp4set2:** Puntos conseguidos por el jugador que comienza en la posición 4 en el segundo set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp5set2:** Puntos conseguidos por el jugador que comienza en la posición 5 en el segundo set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp6set2:** Puntos conseguidos por el jugador que comienza en la posición 6 en el segundo set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp1set3:** Puntos conseguidos por el jugador que comienza en la posición 1 en el tercer set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp2set3:** Puntos conseguidos por el jugador que comienza en la posición 2 en el tercer set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp3set3:** Puntos conseguidos por el jugador que comienza en la posición 3 en el tercer set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp4set3:** Puntos conseguidos por el jugador que comienza en la posición 4 en el tercer set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp5set3:** Puntos conseguidos por el jugador que comienza en la posición 5 en el tercer set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp6set3:** Puntos conseguidos por el jugador que comienza en la posición 6 en el tercer set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp1set4:** Puntos conseguidos por el jugador que comienza en la posición 1 en el cuarto set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp2set4:** Puntos conseguidos por el jugador que comienza en la posición 2 en el cuarto set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp3set4:** Puntos conseguidos por el jugador que comienza en la posición 3 en el cuarto set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp4set4:** Puntos conseguidos por el jugador que comienza en la posición 4 en el cuarto set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp5set4:** Puntos conseguidos por el jugador que comienza en la posición 5 en el cuarto set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp6set4:** Puntos conseguidos por el jugador que comienza en la posición 6 en el cuarto set del equipo posicionado en la derecha y cuyo tipo es *Integer*.

- **puntosp1set5:** Puntos conseguidos por el jugador que comienza en la posición 1 en el quinto set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp2set5:** Puntos conseguidos por el jugador que comienza en la posición 2 en el quinto set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp3set5:** Puntos conseguidos por el jugador que comienza en la posición 3 en el quinto set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp4set5:** Puntos conseguidos por el jugador que comienza en la posición 4 en el quinto set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp5set5:** Puntos conseguidos por el jugador que comienza en la posición 5 en el quinto set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **puntosp6set5:** Puntos conseguidos por el jugador que comienza en la posición 6 en el quinto set del equipo posicionado en la derecha y cuyo tipo es *Integer*.
- **codigopartido:** código del partido del que está hecho el acta siendo una clave foránea (FK) que hace referencia a la tabla partido de manera que un partido forma parte de un acta determinado.

4.2.1. Servidor Web

Para alojar el servidor que servirá de unión entre las aplicaciones móviles y la base de datos se ha desarrollado una aplicación servidora. Se ha empleado el apache tomcat como un contenedor de servlets para poder alojar el proyecto desarrollado en el servidor. Para ello se ha instalado el programa Xampp, que permite gestionar varios tipos de servidores encontrándose ya previamente configurados y listos para usarse. A continuación, se muestra el panel de control del programa Xampp donde pueden verse la gestión de servidores como Apache o Mysql entre otros:

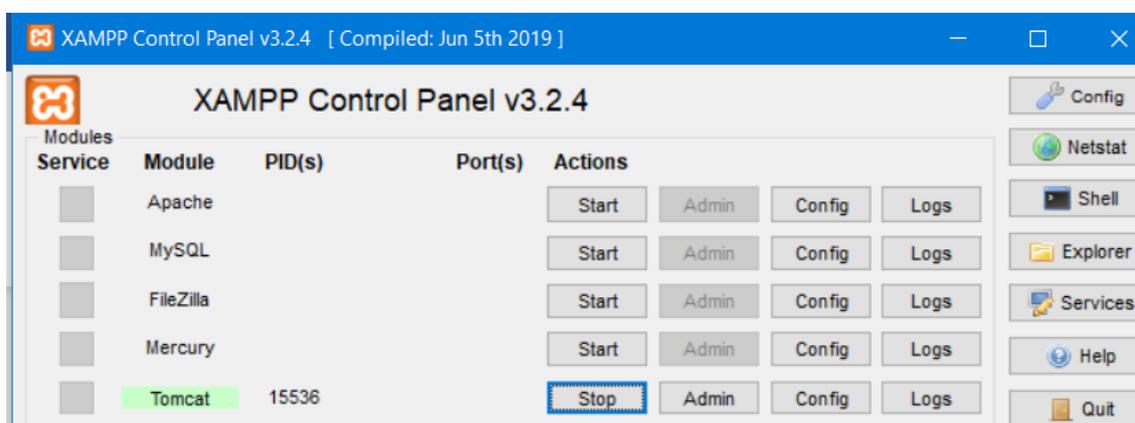


Ilustración 14. Interfaz Xampp

Desde la consola anterior se puede arrancar/parar el servidor Tomcat. Una vez iniciado Tomcat, podrá desplegarse en el servidor la aplicación desarrollada para dicho fin. Para ello se ha desarrollado un proyecto Java, el cual ha sido desarrollado en el entorno de programación Eclipse.

4.2.2. Aplicación servidor

El servidor implementado solo consta de una parte de back-end, ya que la parte front-end no es necesaria para este proyecto. El principal objetivo es que sirva de punto de conexión entre la aplicación Android y la base de datos, desarrollándose servlets para recibir y enviar peticiones desde el móvil.

Para poder comenzar con la creación del servidor, ha sido necesario instalar Maven en eclipse para posteriormente poder convertir el proyecto (que será de tipología *Dynamic Web Project*) a un *Maven Project*. Los proyectos dinámicos pueden contener recursos Java EE tales como servlets, Archivos JSP, filtros y metadatos asociados. Adicionalmente, puede contener recursos estáticos como HTML, archivos e imágenes. Estos proyectos son muy útiles para poder introducir cualquier tipo de información para realizar las conexiones.

Para facilitar la organización del proyecto, este ha sido estructurado tal y como se muestra en la siguiente imagen:

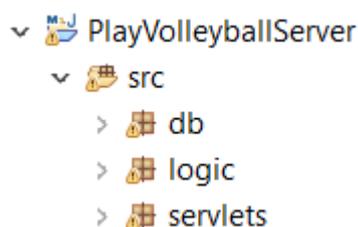


Ilustración 15. Imagen estructura servidor.

A continuación, se detalla el contenido de cada uno de los paquetes en los que se encuentra estructurado el código del proyecto:

db

En este paquete se encuentran los distintos elementos o tablas que contendrá la base de datos, así como la conexión con esta. Contiene las siguientes clases:

- **Acta:** Guarda los campos de la tabla Acta de la base de datos, y sus getter y setter correspondientes.
- **Arbitros:** Guarda los campos de la tabla Arbitros de la base de datos, y sus getter y setter correspondientes.

- **Entrenador:** Guarda los campos de la tabla Entrenador de la base de datos, y sus getter y setter correspondientes.
- **Equipo:** Guarda los campos de la tabla Equipo de la base de datos, y sus getter y setter correspondientes.
- **Jugador:** Guarda los campos de la tabla Jugador de la base de datos, y sus getter y setter correspondientes.
- **Login:** Guarda los campos de la tabla Login de la base de datos, y sus getter y setter correspondientes.
- **Partidos:** Guarda los campos de la tabla Partido de la base de datos, y sus getter y setter correspondientes.
- **ConnectionDDBB:** Esta clase está destinada a la interacción con la base de datos, tanto abrir y cerrar conexión, como las funciones que ejecutan acciones sobre ella. Se emplea esta clase para realizar la conexión con la base de datos, contiene los siguientes métodos:
 - **obtainConnection**(boolean autoCommit): Esta función es la encargada de conectarse con la BBDD, lo intentará 5 veces llamando al método: *DriverManager.getConnection(ddburl, ddbbemail, ddbbpassword)*.
 - **getStatement**(Connection con, String sql): Este método, es el encargado de hacer las llamadas SQL a la BBDD, entrándole como parámetros la conexión y la consulta SQL.
 - **closeTransaction**(Connection con): Función que cierra la conexión con la BBDD haciendo la llamada a la función: *con.commit()*.
 - **cancelTransaction**(Connection con): Método que cancela el envío de la consulta SQL a la base de datos: *con.rollback()*.
 - **closeConnection**(Connection con): Guarda un mensaje en el log en el instante en el que se ha cerrado la conexión con la base de datos.
 - **PreparedStatement** “ ”(Connection con): Se le introduce como parámetro la conexión a la DDBB, dentro se llama a la función *getStatement()*, en la cual se introducirá la consulta SQL si todo ha sido efectuado correctamente, en los archivos de log del servidor (que se encuentran en una carpeta del disco C:):

logic

En este paquete se encuentra la creación de los logs de la aplicación para tener un registro de los datos que se requieran así como de las conexiones. Adicionalmente, se encuentra una clase con toda la lógica principal del programa que contiene todas las operaciones de los datos llamando los diferentes PreparedStatements definidos en la clase ConnectionDDBB().

- **Log:** Se utiliza para la creación de los logs de la aplicación. Contiene tres logs, “mqttlog” para información de las operaciones con mqtt, “dblog” para las operaciones relacionadas con la base de datos y “log” para el resto de las acciones del sistema.
- **Logic:** Contiene la interacción del server con la base de datos, diversos métodos que se llaman en el paquete de servlets y las operaciones con los distintos datos, es decir, contiene la lógica del programa. Métodos contenidos:
 - **getPartidosFromDB():** Devuelve la lista de los partidos existentes en la base de datos para ser llamado posteriormente en un servlet.
 - **getJugador1FromDB():** Devuelve la lista de los jugadores del equipo local existentes en la base de datos para ser llamado posteriormente en un servlet.
 - **getJugador2FromDB():** Devuelve la lista de los jugadores del equipo visitante existentes en la base de datos para ser llamado posteriormente en un servlet.
 - **getActaFromDB():** Devuelve el código del acta existente en la base de datos para ser llamado posteriormente en un servlet.
 - **getApellidosEntrenadorEq1():** Devuelve la lista de los entrenadores del equipo local existentes en la base de datos para ser llamado posteriormente en un servlet.
 - **getApellidosEntrenadorEq2():** Devuelve la lista de los entrenadores del equipo visitante existentes en la base de datos para ser llamado posteriormente en un servlet.
 - **getArbitros():** Devuelve la lista de los árbitros existentes en la base de datos para ser llamado posteriormente en un servlet.
 - **getLicenciaArbitros():** Devuelve la lista con todas las licencias de los árbitros existentes en la base de datos para luego gestionar el login en la aplicación.
 - **getLicenciaEntrenador():** Devuelve la lista con todas las licencias de los entrenadores existentes en la base de datos para luego gestionar el login en la aplicación.

- **getLogin():** Devuelve la lista con todos los logins existentes en la base de datos para ser llamado posteriormente en un servlet.
- **getCapitan1():** Devuelve el apellido del capitán del equipo local.
- **getCapitan2():** Devuelve el apellido del capitán del equipo visitante.
- **getCodigoFromActa():** Devuelve el código del acta existente en la base de datos para ser llamado posteriormente en un servlet.

Servlets

En este paquete se encuentran los servlets creados para que la aplicación pueda acceder a los datos del servidor y recibir información de la app. Para crear un servlet lo primero que hay que hacer es crear un nuevo archivo de tipo servlet en el entorno de programación, se iniciará por defecto con los siguientes métodos: doGet y doPost. Si se quiere enviar información, se debe poner el código en el método doGet, de lo contrario, iría en el doPost. Para enviar datos hay que crear un PrintWriter y almacenar los elementos que se deseen en un ArrayList, esto se hace llamando a una función contenida en la clase Logic la cual contendrá una llamada sql a la base de datos para recibir la información. Tras esto, se debe crear un String json para que almacene los elementos e imprimir el json empleando para ello el PrintWriter anteriormente creado:

- **GetArbitros:** En la función doGet, utiliza la función getArbitros para seleccionar todos los árbitros de la base de datos, pasa la información a formato Json y lo envía a la app.
- **GetCodigoFromActa:** En la función doGet, utiliza la función getCodigoFromActa para seleccionar el código del acta incluido en la base de datos, pasa la información a formato Json y lo envía a la app.
- **GetEntrenadores:** En la función doGet, utiliza la función getApellidosEntrenadorEq1 para seleccionar todos los apellidos de los entrenadores del equipo local incluidos en la base de datos, pasa la información a formato Json y lo envía a la app.
- **GetEntrenadores2:** En la función doGet, utiliza la función getApellidosEntrenadorEq2 para seleccionar todos los apellidos de los entrenadores del equipo visitante incluidos en la base de datos, pasa la información a formato Json y lo envía a la app.
- **GetJugadores:** En la función doGet, utiliza la función getJugador1FromDB para seleccionar todos los apellidos de los jugadores del equipo local incluidos en la base de datos, pasa la información a formato Json y lo envía a la app.
- **GetJugadores2:** En la función doGet, utiliza la función getJugador2FromDB para seleccionar todos los apellidos de los jugadores del equipo visitante incluidos en la base de datos, pasa la información a formato Json y lo envía a la app.

- **GetLicenciaArbitros:** En la función doGet, utiliza la función getLicenciaArbitros para seleccionar todas las licencias de los árbitros que están incluidas en la base de datos, pasa la información a formato Json y lo envía a la app.
- **GetLicenciaEntrenadores:** En la función doGet, utiliza la función getLicenciaEntrenadores para seleccionar todas las licencias de los entrenadores que están incluidas en la base de datos, pasa la información a formato Json y lo envía a la app.
- **GetLogin:** En la función doGet, utiliza la función getLogin para seleccionar todos los registros de usuarios que están incluidos en la base de datos, pasa la información a formato Json y lo envía a la app.
- **GetPartidos:** En la función doGet, utiliza la función getPartidosFromDB para seleccionar todos los partidos que están incluidos en la base de datos, pasa la información a formato Json y lo envía a la app.

Para cada dato a completar de la tabla Acta de la base de datos y para añadir un nuevo registro a la tabla Login, se ha creado un servlet que recibe la información de la aplicación utilizando un PreparedStatement de tipo *insert*, que agrega y modifica los datos en la base de datos. A continuación, se muestra un PreparedStatement creado para la inserción de datos:

```
public static PreparedStatement InsertNewLogin(Connection con) throws SQLException
{
    return getStatement(con, "INSERT INTO login (licencia, contrasena) VALUES (?,?);");
}
```

Ilustración 16. PreparedStatement InsertNewLogin.

4.2.3. MQTT - Mosquitto

En el proyecto se ha escogido un protocolo de comunicación siendo este el MQTT siendo un protocolo ligero y de gran utilidad en casos en los que se requiere un bajo consumo de energía y ancho de banda como es el caso actual.

MQTT [29] son las siglas MQ Telemetry Transport, aunque en primer lugar fue conocido como Message Queing Telemetry Transport. Es un protocolo de comunicación M2M (machine-to-machine) de tipo message queue.

Está basado en la pila TCP/IP como base para la comunicación. En el caso de MQTT cada conexión se mantiene abierta y se "reutiliza" en cada comunicación. Es una diferencia, por ejemplo, a una petición HTTP 1.0 donde cada transmisión se realiza a través de conexión.

Por tanto, el funcionamiento de MQTT es el siguiente:

- Los clientes publican mensajes indicando un *topic*.
- Los clientes pueden suscribirse a uno o varios *topics*. Cuando el *broker* reciba un mensaje con ese tema entonces se encargará de transmitir el mensaje al *subscriber*.
- El *broker* acepta todos los *topics*.

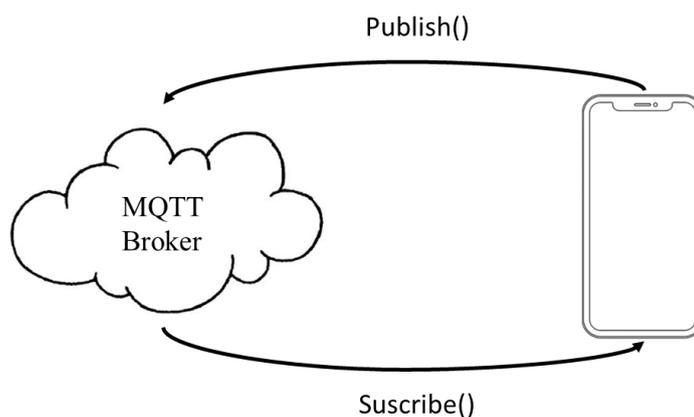


Ilustración 17. Conexión MQTT.

Para la implementación del broker MQTT se ha escogido el servidor Mosquitto. Eclipse Mosquitto™ [30] es un servidor de mensajes de código abierto (con licencia EPL/EDL) que implementa las versiones 3.1 y 3.1.1 del protocolo MQTT. MQTT proporciona un método ligero para llevar a cabo la mensajería utilizando un modelo de publicación/suscripción. Esto lo hace adecuado para la «Internet de los objetos» de mensajería como con sensores de baja potencia o dispositivos móviles como teléfonos, ordenadores embebidos o microcontroladores como el Arduino.

Mosquitto es un broker MQTT OpenSource ampliamente utilizado debido a su ligereza lo que nos permite, fácilmente, emplearlo en gran número de ambientes, incluso si éstos son de pocos recursos.

Para comprobar la conexión MQTT, se ha usado la aplicación Mosquitto Explorer, que enseña todos los topics a los que se suscribe la aplicación, siendo su estructura la siguiente:

```

▼ 8212204 = 8212204
puntos1 = 3
puntos2 = 6
set1 = 0
set2 = 0
horain = 16:51
saque2 = true
saque1 = false
capitan1 = N° 9 Escudero
capitan2 = N° 10 Pastrana
jugadores1 = [N° 3 Contreras,N° 5 Pereira,N° 7 Guillen,N° 9 Escudero,N° 11 Casas,N° 12 ...
jugadores2 = [N° 2 Roca,N° 3 Santamaria,N° 4 Salazar,N° 6 Aranda,N° 9 Ramos,N° 10 Pas..
libero1 = [N° 7 Guillen, N° 11 Casas]
libero2 = [N° 11 Yelamos]
e6 = 3
e5 = 2
e4 = 1
e3 = 6
e2 = 5
e1 = 12
p6 = 9
p5 = 8
p4 = 7
p3 = 12
p2 = 11
p1 = 10
numerocambio = 12
puntoscambio = 2 3
tiempo1equipo1set1 = 3 6
tiempo1equipo2set1 = 6 3
numerosancion = C
equiposancion = A
puntosancion = 1 6 3

```

Ilustración 18. Estructura Mosquitto.

Los topics utilizados por el proyecto son los siguientes:

- **Código:** El topic principal que engloba todo el resto de topics. Se trata del código del partido, de esta forma, si se modifican varios partidos los datos no se sobrescriben. En la ilustración 17 dicho topic se corresponde con el código 8212204.
- **Puntos1:** Este topic se corresponde con los puntos obtenidos durante los sets por el equipo que del lado izquierdo. Se actualizarán conforme el árbitro seleccione un punto.
- **Puntos2:** Este topic se corresponde con los puntos obtenidos durante los sets por el equipo del lado derecho. Se actualizarán conforme el árbitro seleccione un punto.
- **Set1:** Este topic se corresponde con los sets obtenidos durante el partido por el equipo del lado izquierdo. Se actualizarán conforme se obtengan 25 o 15 puntos cuando corresponda.
- **Set2:** Este topic se corresponde con los sets obtenidos durante el partido por el equipo del lado derecho. Se actualizarán conforme se obtengan 25 o 15 puntos cuando corresponda.
- **Horain:** Se corresponde con la hora a la que ha comenzado el partido.
- **Saque1:** Topic que indica true si tiene el saque el equipo del lado izquierdo, en caso contrario indica false.
- **Saque2:** Topic que indica true si tiene el saque el equipo del lado derecho, en caso contrario indica false.
- **Capitan1:** Topic que indica el apellido del capitán del equipo local.
- **Capitan2:** Topic que indica el apellido del capitán del equipo visitante.
- **Jugadores1:** Topic que indica la lista con todos los jugadores del equipo local que disputan el partido.
- **Jugadores2:** Topic que indica la lista con todos los jugadores del equipo visitante que disputan el partido.
- **Libero1:** Topic que indica el apellido de los líberos del equipo local.
- **Libero2:** Topic que indica el apellido de los líberos del equipo visitante.
- **E1,E2,E3,E4,E5,E6:** Topic que indica el número que se encuentra en cada una de las posiciones del equipo que se encuentra en la izquierda del campo.
- **P1,P2,P3,P4,P5,P6:** Topic que indica el número que se encuentra en cada una de las posiciones del equipo que se encuentra a la derecha del campo.
- **Numerocambio:** Topic que indica el número por el que se realiza la sustitución.
- **PuntosCambio:** Topic que indica los puntos en los que se realiza la sustitución.
- **Tiempos1EqXSetX:** Topic que indica la puntuación en la que se ha realizado el primer tiempo del set.

- **Tiempos2EqXsetX:** Topic que indica la puntuación en la que se ha realizado el segundo tiempo del set.
- **Numerosancion:** Topic que indica a quien va dirigida la sanción.
- **Equiposancion:** Topic que indica a que equipo va dirigida la sanción.
- **PuntosSancion:** Topic que indica el set y los puntos en los que se ha realizado la sanción.
- **E1inSetX,E2inSetX,E3inSetX,E4inSetX,E5inSetX,E6inSetX:** Topic que indica la rotación inicial del equipo del lado izquierdo en un set determinado.
- **P1inSetX,P2inSetX,P3inSetX,P4inSetX,P5inSetX,P6inSetX:** Topic que indica la rotación inicial del equipo del lado derecho en un set determinado.
- **Saque1in:** Indica que el equipo que comienza en el lado izquierdo tiene el saque o no.
- **Saque2in:** Indica que el equipo que comienza en el lado derecho tiene el saque o no.
- **Puntos1SetX:** Indica los puntos totales conseguidos por el equipo que comienza en el lado izquierdo en un set determinado.
- **Puntos2SetX:** Indica los puntos totales conseguidos por el equipo que comienza en el lado derecho en un set determinado.
- **HorafinSetX:** Topic que indica la hora a la que se ha terminado un set concreto.
- **Puntose1setXconseguidos, Puntose2setXconseguidos, Puntose3setXconseguidos, Puntose4setXconseguidos, Puntose5setXconseguidos, Puntose6setXconseguidos:** Puntos conseguidos por la rotación inicial del equipo del lado izquierdo en un set determinado.
- **Puntosp1setXconseguidos, Puntosp2setXconseguidos, Puntosp3setXconseguidos, Puntosp4setXconseguidos, Puntosp5setXconseguidos, Puntosp6setXconseguidos:** Puntos conseguidos por la rotación inicial del equipo del lado derecho en un set determinado.

4.2.4. Aplicación móvil

El desarrollo de la aplicación se ha realizado mediante Android Studio con Java como lenguaje de programación. A continuación, se muestra un pequeño esquema con la estructura de las diferentes pantallas de la aplicación:

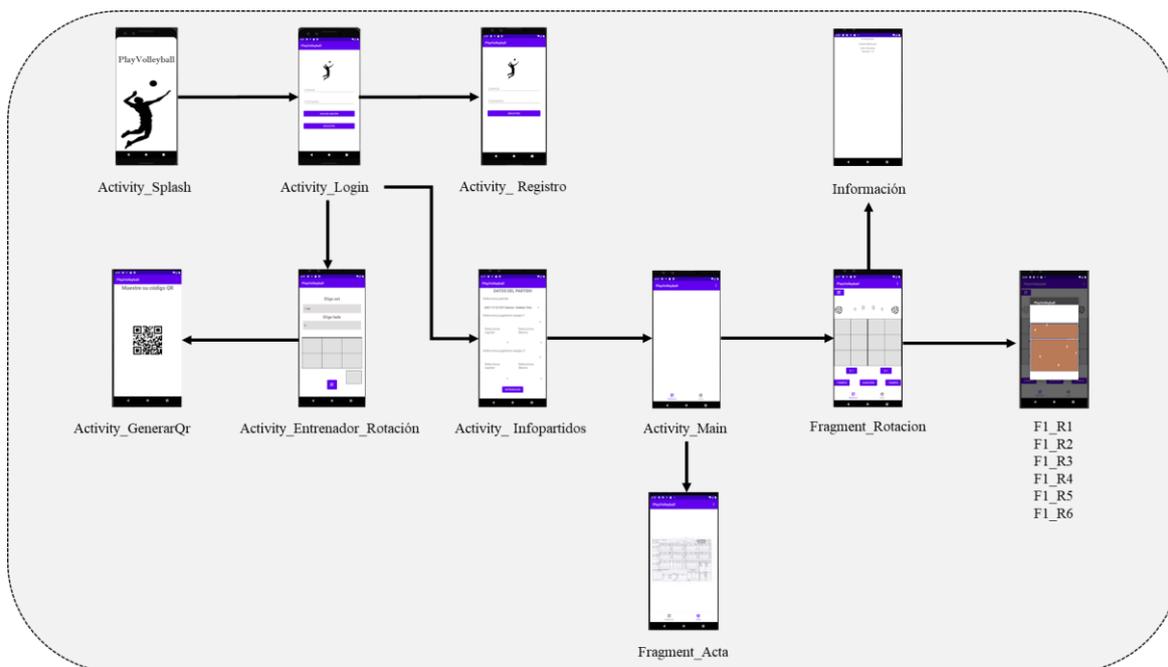


Ilustración 19. Diagrama de pantallas de la aplicación

La aplicación se ha realizado en dos idiomas, español e inglés, de esta forma si el usuario tiene su móvil en inglés, los datos aparecerán en inglés y si el usuario tiene el móvil en español, la información saldrá en español.

LAYOUTS

- **Activity_Splash:** Pantalla inicial en la que se muestra una animación y aparece la imagen principal de la aplicación. Tras esto, aparece la pantalla de login.
- **Activity_Login:** Pantalla en la que se introduce la licencia del entrenador o del árbitro, y una contraseña para acceder a la aplicación. En el caso de que no se esté registrado se accede a la pantalla de registro pinchando en el botón de registrar. El login y el registro se ha realizado comprobando las licencias existentes en la base de datos. Al utilizar una pantalla de login/registro, se mejora la seguridad de la aplicación para prevenir accesos indebidos a los datos de los partidos.
- **Activity_Registro:** En esta pantalla se muestra la funcionalidad de registro mediante una de las licencias registradas en la base de datos y una contraseña que debe ser mayor o igual a 6 caracteres para tener una contraseña segura. Una vez realizado el registro, se pasa a la pantalla de login.
- **Activity_Entrenador_Rotacion:** Pantalla a la que solo acceden los entrenadores, donde introducen la rotación de cada set y el lado del campo en el que empiezan a jugar. A partir de aquí se genera un código QR que posteriormente será leído por los árbitros.

- **Activity_GenerarQR:** Pantalla en la que se muestra la imagen del código QR generado por los entrenadores.
- **Activity_InfoPartidos:** Pantalla en la que se selecciona el partido que se va a jugar y los jugadores de cada equipo que van a jugar dicho partido. A partir de los jugadores seleccionados, se selecciona al capitán y a los líberos. Después, se accede al Main.
- **Activity_Main:** En esta pantalla se muestra el navigation menu por el que se podrá ir cambiando de pestañas. Se ha creado mediante un fragmento para prevenir que se superponga al navigation menu. a su vez hay implementado un menú el que se puede seleccionar la información de la aplicación o cerrar sesión y volver a la pestaña del login.
- **Fragment_Rotacion:** Pantalla donde se realiza todo el partido. Se encuentra una tabla que asemeja a las posiciones de la pista, que se rellenaran mediante el escaneo del QR accediendo mediante el botón de la esquina superior izquierda. A partir de ahí, se puede seleccionar quien tiene el saque inicial tras haber realizado el sorteo y las posiciones de la pista teniendo dos opciones, punto o sustitución. Existe un botón llamado 5-1 que muestra las rotaciones de dicho sistema a partir de introducir el número del colocador. También están los botones de tiempo para solicitar hasta un máximo de dos tiempos muertos y sanción para añadir las sanciones necesarias, a quien va dirigido (si entrenador o jugador) y a que equipo.
- **F1_R1, F1_R2, F1_R3, F1_R4, F1_R5, F1_R6:** Pantallas que se generan mediante una ventana emergente, en el que se introduce el número del colocador para saber en qué posición se encuentra y desplegar la rotación correspondiente.
- **Fragment_Acta:** Pantalla que muestra el acta completada en tiempo real, mediante la librería PDFView. Para poder completar el acta, se ha utilizado la librería itext7, en donde se añade una imagen de un acta oficial y se crean tablas añadiendo la información encima. A su vez, cada vez que se accede a esta pantalla se descarga el acta en pdf en el dispositivo local.
- **Información:** En esta pantalla se muestra el nombre del creador de la aplicación con la versión en la que está. Esta pantalla se actualizará conforme salgan más actualizaciones en un futuro. Se accede a ella mediante el menú principal (los tres puntos que aparecen en la esquina superior derecha).

En la carpeta **mimap** se encuentra la imagen que será icono de la aplicación en diferentes tamaños.

En la carpeta **values** se encuentran los dos archivos strings para los dos idiomas indicados anteriormente.

Actividades

Las actividades creadas en la aplicación se dividen en:

- **Data:** En esta carpeta se muestran las clases creadas también en el servidor.
 - **Arbitro:** Se implementa la clase arbitro con sus parámetros.
 - **Entrenadores:** Se implementa la clase entrenadores con sus parámetros.
 - **Equipo:** Se implementa la clase equipo con sus parámetros.
 - **Login:** Se implementa la clase login con sus parámetros.
 - **Partidos:** Se implementa la clase partidos con sus parámetros

- **Task:** En esta carpeta se crean todas las tareas asíncronas a las que se llamarán mediante las diferentes actividades.
 - **TaskSelectArbitros:** Tarea que llama a la actividad InfoPartidosActivity. En el método OnPostExecute se llamará cuando la tarea esté terminada, cambiando los datos conforme a los árbitros recibidos.
 - **TaskSelectEntrenador1** y **TaskSelectEntrenador2:** Tarea que llama a la actividad InfoPartidosActivity. En el método OnPostExecute se llamará cuando la tarea esté terminada, cambiando los datos conforme a los entrenadores recibidos.
 - **TaskSelectJugadores** y **TaskSelectJugadores2:** Tarea que llama a la actividad InfoPartidosActivity. En el método OnPostExecute se llamará cuando la tarea esté terminada, cambiando los datos conforme a los jugadores recibidos.
 - **TaskSelectJugadoresRotacion** y **TaskSelectJugadores2Rotacion:** Tarea que llama a la actividad RotacionFragment. En el método OnPostExecute se llamará cuando la tarea esté terminada, cambiando los datos conforme a los jugadores recibidos.
 - **TaskSelectListArbitros:** Tarea que llama a la actividad RegisActivity. En el método OnPostExecute se llamará cuando la tarea esté terminada, cambiando los datos conforme a los árbitros recibidos.
 - **TaskSelectListArbitrosLogin:** Tarea que llama a la actividad LoginActivity. En el método OnPostExecute se llamará cuando la tarea esté terminada, cambiando los datos conforme a las licencias de los árbitros recibidos.
 - **TaskSelectListEntrenador:** Tarea que llama a la actividad RegisActivity. En el método OnPostExecute se llamará cuando la tarea esté terminada, cambiando los datos conforme a los entrenadores recibidos.

- **TaskSelectListEntrenadorLogin:** Tarea que llama a la actividad LoginActivity. En el método onPostExecute se llamará cuando la tarea esté terminada, cambiando los datos conforme a las licencias de los entrenadores recibidos.
- **TaskSelectLogin:** Tarea que llama a la actividad LoginActivity. En el método onPostExecute se llamará cuando la tarea esté terminada, cambiando los datos conforme a los login recibidos.
- **TaskSelectPartidos:** Tarea que llama a la actividad InfoPartidosActivity. En el método onPostExecute se llamará cuando la tarea esté terminada, cambiando los datos conforme a los partidos recibidos.
- **Llamadas:** En este apartado se encuentran todas las actividades. Son la parte más importante de la aplicación ya que se encuentra la implementación principal de todas las pantallas.
 - **ActaFragment:** Fragment en el que se recibe toda la información necesaria para completar el acta. Se crea una nueva variable por cada dato que se va a escribir en el acta. Para poder escribir en el acta, se ha tenido que crear una tabla para cada dato, creando un tamaño de columna determinado. Tras esto, se establecen los márgenes, y se añaden nuevas celdas y párrafos. Para cada dato, se ha tenido que establecer un margen determinado para cuadrar el dato encima del acta en la posición correcta. Como se ha explicado anteriormente, se ha usado PDFView para mostrar el acta en PDF en la pantalla.

```
float[] columnWidth1 = {720, 220, 220, 100};
```

Ilustración 20. Código para establecer el tamaño de la columna.

```
table6.setMargins( marginTop: -15, marginRight: 0, marginBottom: 0, marginLeft: 890);

table6.addCell(new Cell().setBorder(Border.NO_BORDER).add(new Paragraph( text: "A").setFontSize(20).setBorder(Border.NO_BORDER)));
table6.addCell(new Cell().setBorder(Border.NO_BORDER).add(new Paragraph(eq1).setFontSize(20).setBorder(Border.NO_BORDER)));
table6.addCell(new Cell().setBorder(Border.NO_BORDER).add(new Paragraph(eq2).setFontSize(20).setBorder(Border.NO_BORDER)));

table6.addCell(new Cell().setBorder(Border.NO_BORDER).add(new Paragraph( text: "B").setFontSize(20).setBorder(Border.NO_BORDER)));

if (division.equals("femenina") && categoria.equals("senior")) {
    table4.setMargins( marginTop: -20, marginRight: 0, marginBottom: 0, marginLeft: 350);
    table4.addCell(new Cell().setBorder(Border.NO_BORDER).add(new Paragraph( text: "X").setFontSize(20).setBorder(Border.NO_BORDER)));
    table4.addCell(new Cell().setBorder(Border.NO_BORDER).add(new Paragraph( text: "X").setFontSize(20).setBorder(Border.NO_BORDER)));
}
```

Ilustración 21. Creación de las tablas, párrafos y celdas.

```
document.add(image.setFixedPosition( left: 0, bottom: 0));
document.add(table1);
document.add(table2);
document.add(table3);
```

Ilustración 22. Añadir imagen y tablas al PDF.

```
PDFView pdfView = (PDFView) view.findViewById(R.id.acta);
pdfView.fromFile(file).load();
```

Ilustración 23. Código para la lectura del PDF.

```
public void createPDF() throws FileNotFoundException {

    OutputStream outputStream = new FileOutputStream(file);

    PdfWriter writer = new PdfWriter(file);
    PdfDocument pdfDocument = new PdfDocument(writer);
    Document document = new Document(pdfDocument, PageSize.A1.rotate());
    document.setMargins( topMargin: 0, rightMargin: 0, bottomMargin: 0, leftMargin: 0);
    Drawable d1 = ContextCompat.getDrawable(getActivity(), R.drawable.acta);
    Bitmap bitmap1 = ((BitmapDrawable) d1).getBitmap();
    ByteArrayOutputStream stream1 = new ByteArrayOutputStream();
    bitmap1.compress(Bitmap.CompressFormat.PNG, quality: 100, stream1);
    byte[] bitmapData1 = stream1.toByteArray();
    ImageData imageData = ImageDataFactory.create(bitmapData1);
    Image image = new Image(imageData);
```

Ilustración 24. Código para escribir en el acta en PDF.

- **EntrenadorActivity**: Fragment donde el entrenador introduce los datos relacionados con la rotación del partido en cada set. Posteriormente, se le pasa la información introducida en formato bitmap para generar el código QR.

```

Bitmap bitmap = QRCode.from(String.valueOf(data)).bitmap();

Intent intent = new Intent( packageContext: EntrenadorActivity.this, QrGeneradoActivity.class);
intent.putExtra( name: "bitmap", bitmap);

```

Ilustración 25. Pasar información código QR.

- **F1_r1 y F1_r1_eq2:** Clase que genera la primera rotación del sistema 5-1 con los datos del QR.
- **F1_r2 y F1_r2_eq2:** Clase que genera la segunda rotación del sistema 5-1 con los datos del QR.
- **F1_r3 y F1_r3_eq2:** Clase que genera la tercera rotación del sistema 5-1 con los datos del QR.
- **F1_r4 y F1_r4_eq2:** Clase que genera la cuarta rotación del sistema 5-1 con los datos del QR.
- **F1_r5 y F1_r5_eq2:** Clase que genera la quinta rotación del sistema 5-1 con los datos del QR.
- **F1_r6 y F1_r6_eq2:** Clase que genera la sexta rotación del sistema 5-1 con los datos del QR.
- **Info:** Clase que solo llama al layout de la información.
- **InfoPartidosActivity:** Clase donde se obtiene toda la información relacionada con los partidos y los jugadores. Toda la información se despliega en listas, donde luego el árbitro selecciona la información necesaria. Para recibir la información necesaria se llama a las tareas correspondientes. A continuación se muestra un ejemplo:

```

private void loadPartidos() {
    new TaskSelectPartidos( activity: this).execute("http://192.168.65.1:8080/PlayVolleyballServer/GetPartidos");
}

```

Ilustración 26. Llamada a la tarea TaskSelectPartidos.

- **InfoRotacion:** Clase que recoge información para luego mandarla al acta, creando un constructor con la información y los getter y setter correspondientes.
- **LoginActivity:** Clase que obtiene todas las licencias de los árbitros y los entrenadores. Si las credenciales son de un entrenador, se pasará a

EntrenadorActivity, en caso de que sean las credenciales de un árbitro, se accede a la clase InfoPartidosActivity.

- **MainActivity**: esta es la actividad principal que implementa la navigation bar siendo el encargado de llamar a todos los fragment para que cuando se pulse en el icono deseado lleve a la pantalla correcta y no se sobreponga la pantalla al menú. También llama al menú que se encuentra en la esquina superior derecha comprobando que si la opción elegida es info, lleve a la pantalla de información y si pulsa la otra opción, lleve a la pantalla de login cerrando la sesión.
- **QrGeneradoActivity**: Clase que genera el icono del código QR a partir del bitmap recibido.
- **RegisActivity**: Clase donde se registra a los usuarios en la base de datos. Comprueba si la licencia introducida corresponde con la de un árbitro o la de un entrenador, a partir de ahí genera una AsyncTask que envía la información al servidor.
- **SplashActivity**: Clase que establece una animación que va a tener una duración de 3000 ms. Tras esto se llama a la pantalla de login.
- **RotacionFragment**: Clase donde se implementa la funcionalidad principal del programa. Para las sanciones se genera una ventana emergente donde se introduce a quien va dirigida la sanción y a que equipo. En los botones de los sistemas de rotación, se genera una ventana emergente donde se introduce la posición del colocador, a partir de ahí, se abre otra ventana emergente con la imagen de la rotación pertinente. Respecto al botón para leer el QR, se genera un IntentIntegrator que abre la cámara del dispositivo. Para el saque, se establece un botón con imagen, que en cuanto se pulse el otro botón de saque desaparece. Para cada una de las posiciones en el campo, aparece un menú desplegable donde aparecen dos opciones, la primera es para marcar un punto, a partir de ahí se genera todo el código para sumar el punto al marcador y en caso de que no tuviesen el saque, se genere una rotación de posiciones. La otra opción es realizar una sustitución, generando una ventana emergente donde se introduce el numero por el que se sustituye al jugador, a partir de ahí la posición se sustituye por dicho número introducido. Para enviar toda la información a la base de datos, se crean AsyncTasks llamando a la queries correspondientes. A continuación, se muestra un ejemplo:

```

public String sendData40(Boolean puntos1) {
    try {
        RequestBody formBody = new FormBody.Builder()
            .add( name: "Puntos1 ", String.valueOf(puntos1))

            .build();
        Log.i( tag: "SendData", msg: "puntos1 " + puntos1);
        Request request = new Request.Builder()
            .url("http://192.168.65.1:8080/PlayVolleyballServer/RecibirSaque2In?saque2in=" + puntos1)
            .build();

        Response response = client.newCall(request).execute();
        return response.body().string();
    } catch (IOException e) {
        return "Error: " + e.getMessage();
    }
}
}

```

Ilustración 27. Insertar datos en la bbdd.

- **MQTT:** Para recibir la información en tiempo real y que pueda haber varios árbitros conectados a un mismo partido, se ha creado una conexión MQTT en la aplicación. Para cada dato que se modifica durante un partido, se realiza un publish al MQTT mediante el código del partido. Para obtener la información, la aplicación se suscribe a los topics necesarios. Esta conexión se realiza tanto en la clase InfoRotacion como en el fragmento RotacionFragment.

```

try {
    IMqttToken token = client2.connect();
    token.setActionCallback(new IMqttActionListener() {
        @Override
        public void onSuccess(IMqttToken asyncActionToken) {
            //If the connection is ok
            Log.i( tag: "MQTT", msg: "MQTT connected");
            //Suscribe the topics

            suscripcionTopics( capitan1: codigo + "/" + "0");
            suscripcionTopics( capitan1: codigo + "/" + "");

            MqttMessage msg = new MqttMessage("0".getBytes());
            MqttMessage msg1 = new MqttMessage("").getBytes());

            MqttMessage msg2 = new MqttMessage(str.getBytes());

            try {
                client2.publish( topic: codigo + "/puntos1", msg);
                client2.publish( topic: codigo + "/puntos2", msg);
                client2.publish( topic: codigo + "/set1", msg);
                client2.publish( topic: codigo + "/set2", msg);
            }
        }
    });
}

```

Ilustración 28. Publish MQTT.

```

//MQTT topics to suscribe the application
private void suscripcionTopics(String saque1) {
    try {
        Log.i( tag: "main", msg: "info = " + saque1);
        //TODO suscribir a todos los topics de los sensores

        client2.subscribe( topic: codigo + "/saque1", qos: 0);
    }
}

```

Ilustración 29. Suscripción a los topics.

En el androidmanifest se han introducido todas las actividades que van a mostrar pantallas y también permisos para que se puedan implementar diferentes métodos como la conexión a internet, leer y escribir el almacenamiento o leer el estado del teléfono.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.playvolleyball">
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission
        android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.WAKE_LOCK" />
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.VIBRATE" />
```

Ilustración 30. Android Manifest

En el gradle.app se han introducido todas las dependencias necesarias para que funciones los métodos tales como las de leer y escribir el PDF o la conexión con MQTT.

También hubo que poner una versión anterior del Android para que funcionasen las AsyncTask.

```
implementation 'com.android.support:support-v13:29.0.2'
implementation 'com.android.support:appcompat-v7:29.0.0'
implementation 'com.android.support:preference-v7:29.0.0'
implementation 'com.google.android.material:material:1.0.0'
implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
implementation 'androidx.lifecycle:lifecycle-extensions:2.0.0'
implementation 'androidx.legacy:legacy-support-v4:1.0.0'
//leer pdf
implementation 'com.github.barteksc:android-pdf-viewer:2.8.2'
implementation 'org.apache.commons:commons-io:1.3.2'

implementation 'com.itextpdf:itext7-core:7.1.3'
//generar qr
implementation 'androidmads.library.qrgenerator:QRGenerator:1.0.4'
implementation 'com.journeyapps:zxing-android-embedded:3.5.0'
implementation 'com.github.kenglxn.QRGen:android:2.5.0'
implementation 'androidx.lifecycle:lifecycle-viewmodel-savedstate:1.0.0-alpha03'
implementation 'com.android.support:support-v4:29.0.0'
implementation files('libs/postgresql-42.2.1.jre7.jar')

implementation 'org.eclipse.paho:org.eclipse.paho.client.mqttv3:1.2.1'

implementation('org.eclipse.paho:org.eclipse.paho.android.service:1.1.1') {
    exclude group: 'com.android.support'
    exclude module: 'appcompat-v7'
    exclude module: 'support-v4'
}

testImplementation 'junit:junit:4.+'
androidTestImplementation 'androidx.test.ext:junit:1.1.2'
androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
```

Ilustración 31. Gradle App

4.3. Testimonios de árbitros

Para cubrir todas las necesidades arbitrales en la aplicación, se ha realizado una entrevista a algunos árbitros y jugadoras con el fin de tener un software completo. A continuación, se muestran algunos testimonios:

Diana González como árbitro nivel II: “Crear una aplicación donde se incluya escribir las observaciones del acta mediante voz, agilizando el proceso y evitando pérdidas de tiempo”.

Laura Viñuelas como árbitro nivel I: “Digitalizar el acta de una manera más sencilla sin realizar cursos previos”.

Sandra Barroso como jugadora: “Implementar los sistemas de formación para que los árbitros sepan detectarlo más rápido durante los encuentros”.

Mario Bernaola CTNA (desarrollador de una aplicación para hacer el acta con el móvil): “La aplicación engloba muchas otras aplicaciones en una sola, lo que la hace mucho más completa.”

5. Futuras mejoras

El proyecto implementado cumple todos los requisitos explicados en el Objetivo del proyecto. Aun así, existen múltiples mejoras que se pueden llevar a cabo para mejorar la aplicación:

1. Realizar estadísticas de los jugadores e implementarlas en la pantalla de los entrenadores para que tengan acceso.
2. Implementar las observaciones mediante voz usando el micrófono del teléfono.
3. Implementar el sorteo inicial del partido en la aplicación.
4. Mejorar el diseño de la pantalla de selección de jugadores.
5. Implementar un método que se pueda realizar la firma del acta de manera digital.
6. Implementar la aplicación para otros sistemas operativos, tal como IOS.
7. Implementar un método para poder imprimir el acta.
8. Implementar un método para poder enviar el acta por correo electrónico.
9. Usar otros protocolos de seguridad para proteger la información de los datos.
10. Implementar la aplicación para vóley playa.
11. Implementar un método que envíe la información del partido en tiempo real a la Federación.

6. Conclusiones

Este Trabajo de Fin de Grado ha conseguido subsanar una carencia en los procesos arbitrales. La autora de este proyecto ha ganado grandes conocimientos sobre Android Studio, tales como implementar y leer un código QR así como leer y escribir un PDF.

En concreto, la autora de este proyecto va a utilizar la aplicación en su proceso arbitral adaptándolo a la base de datos correspondiente, facilitándole ver las faltas de rotación en tiempo real.

Actualmente, cualquier persona tiene un smartphone en su vida, lo que significa que una aplicación móvil es muy útil y facilita la vida de las personas.

A parte de todos los conocimientos teóricos y prácticos adquiridos durante la carrera de Ingeniería Informática, este proyecto ha dado lugar a aprender y desarrollar una aplicación Android muy completa.

7. Coste del proyecto

El presupuesto para el proyecto mostrado en este documento se divide en 3 bloques: coste de personal, coste de material y gastos generales. A continuación, se desglosan cada uno de estos costes.

7.1. Coste de personal

CONCEPTO	HORAS	COSTE/HORA	COSTE TOTAL
Jefe del proyecto	40	22€	880€
Analista	60	18€	1080€
Programador	200	12€	2400€
Total			4360€

Tabla 9. Coste de personal de ingeniería.

CONCEPTO	HORAS	COSTE/HORA	COSTE TOTAL
Documentalista	160€	10€	1600€
Total			1600€

Ilustración 32. Coste del personal de secretaría.

DESCRIPCIÓN	TOTAL
Coste Personal Ingeniería	4360€
Coste Personal Secretaría	1600€
Total	5960€

Ilustración 33. Coste total personal.

El coste de personal se encuentra dividido en:

- Costes de Ingeniería
 - Desarrollo del prototipo
 - Gestión del proyecto
 - Análisis asociado al desarrollo del proyecto
 - Pruebas del código
- Costes de Secretaría:
 - Desarrollo de la documentación

7.2. Coste de material

El coste del material necesario para la realización de este proyecto se divide en 2 apartados:

DESCRIPCIÓN	NÚMERO	COSTE UNITARIO	TOTAL
Ordenador Lenovo Legion Y720 con Windows 10	1	1.769€	1769€
Monitor Lenovo L24i-30 23,8"	1	270€	270€
Móvil Huawei P10	1	195€ actualmente	195€
Móvil Xiaomi Redmi 8	1	160€ actualmente	160€
Disco duro Lenovo 1TB	1	130€	130€
Total			2524€

Tabla 10. Coste Hardware

DESCRIPCIÓN	NÚMERO	COSTE UNITARIO	TOTAL
Android Studio	1	0€	0€
Java	1	0€	0€
PostgreSQL	1	0€	0€
Apache Tomcat	1	0€	0€
Xampp	1	0€	0€
Insomnia	1	0€	0€
Eclipse	1	0€	0€
Microsoft Office (Licencia gratuita UAH)	1	0€	0€
MQTT Explorer	1	0€	0€
Total			0€

Tabla 11. Coste Software

DESCRIPCIÓN	TOTAL
Coste Hardware	2524€
Coste Software	0€
Total	2524€

Tabla 12. Coste Software

7.3. Costes Generales

En este tipo de gasto se incluyen gastos variables como el material de oficina, costes de dietas, desplazamientos. Se estima un gasto en concepto de gastos generales de un 15% de los costes de personal y material.

Costes Generales	1273€
-------------------------	--------------

Tabla 13. Costes Generales

7.4. Coste Total

DESCRIPCIÓN	TOTAL
Costes Personal	5960€
Costes Material	2524€
Costes Generales	1273€
Total	9757€

Tabla 14. Coste Total

El coste total del proyecto asciende a la cantidad de NUEVE MIL SETECIENTOS CINCUENTA Y SIETE EUROS.

8. Bibliografía

- [1] E. Álvarez, «11 inventos tecnológicos que revolucionaron el deporte,» [En línea]. Available: <https://computerhoy.com/listas/life/11-inventos-tecnologicos-que-revolucionaron-deporte-44950>.
- [2] P. Arnau, «El impacto de las aplicaciones sobre deportes,» [En línea]. Available: <https://deportesinc.com/numeros-y-estadisticas/aplicaciones-sobre-deportes/>.
- [3] F. I. d. Voleibol, «GUIAS e instrucciones de arbitraje,» [En línea]. Available: https://www.rfevb.com/Files/Descargas/FIVB_Guias_e_Instrucciones_de_Arbitraje_2020-pdfEs20200519103110.pdf.
- [4] «Árbitros de voleibol: Gestos y señales,» [En línea]. Available: <https://esvoley.com/arbitros-de-voleibol-senales-y-gestos/>.
- [5] E. d. C. M. Maldavsky, «Sistemas de juego de Voleibol,» [En línea]. Available: <https://starsvolleyball.es/sin-categoria/sistemas-de-juego-voleibol/>.
- [6] GuillaumeVT, «Volleyball Referee App Dowload,» [En línea]. Available: <https://play.google.com/store/apps/details?id=com.tonkar.volleyballreferee>.
- [7] U. p. d. l. a. AppLineUp. [En línea]. Available: <https://www.rfevb.com/n/descarga-la-aplicacion-applineup-para-su-uso-en-los-campeonatos-de-espana>.
- [8] J. R. D. -. Youtube, «Como leer un código QR en Android Studio,» [En línea]. Available: <https://www.youtube.com/watch?v=ststFs4x4p4-->.
- [9] E.-S. A. Download. [En línea]. Available: <https://www.dataproject.com/Products/global/en/volleyball/eScoresheet>.
- [10] D. V. Essential. [En línea]. Available: <https://www.dataproject.com/Products/EN/en/Pallavolo/DataVideo>.
- [11] DataVolley. [En línea]. Available: <https://www.dataproject.com/Products/EN/en/Volleyball/DataVolley4>.
- [12] J. D. M. Pozo, «Entornos de programación móviles,» [En línea]. Available: [https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_\(Modulo_3\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_(Modulo_3).pdf).
- [13] A. Casans, «Las mejores herramientas de desarrollo de apps moviles,» [En línea]. Available: <https://www.hiberus.com/crecemos-contigo/mejores-herramientas-de-desarrollo-de-apps-movil/>.
- [14] Wikipedia, «Apache Cordova,» [En línea]. Available: https://es.wikipedia.org/wiki/Apache_Cordova.
- [15] Wikipedia, «PhoneGap,» [En línea]. Available: <https://es.wikipedia.org/wiki/PhoneGap>.
- [16] Wikipedia, «Flutter,» [En línea]. Available: [https://es.wikipedia.org/wiki/Flutter_\(software\)](https://es.wikipedia.org/wiki/Flutter_(software)).
- [17] «En el desarrollo de apps ¿Flutter o Ionic?,» [En línea]. Available: <https://www.syntonize.com/flutter-o-ionic/>.
- [18] Wikipedia, «Bases de datos,» [En línea]. Available: https://es.wikipedia.org/wiki/Base_de_datos.
- [19] Wikipedia, «MySQL,» [En línea]. Available: <https://es.wikipedia.org/wiki/MySQL>.
- [20] Wikipedia, «PostgreSQL,» [En línea]. Available: <https://es.wikiped>

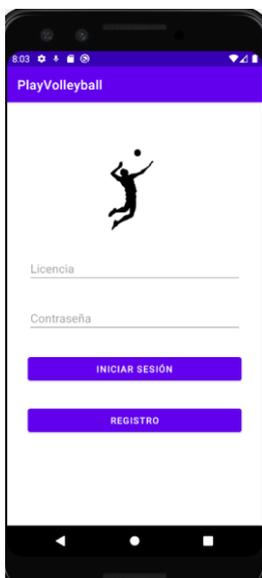
ia.org/wiki/PostgreSQL.

- [21] V. d. Guiadev, «PostgreSQL vs MongoDB,» [En línea]. Available: <https://guiadev.com/postgresql-vs-mongodb/>.
- [22] Wikipedia, «MariaDB,» [En línea]. Available: <https://es.wikipedia.org/wiki/MariaDB>.
- [23] Wikipedia, «Tomcat,» [En línea]. Available: <https://es.wikipedia.org/wiki/Tomcat>.
- [24] L. Gracia, «Tomcat 7: Nuevas características,» [En línea]. Available: <https://unpocodejava.com/2010/08/09/tomcat-7-nuevas-caracteristicas/>.
- [25] G. B, «¿Qué es NGINX y cómo funciona?,» [En línea]. Available: <https://www.hostinger.mx/tutoriales/que-es-nginx>.
- [26] Á. d. León, «Servidor Litespeed,» [En línea]. Available: <https://blog.infranetworking.com/servidor-litespeed/>.
- [27] «XAMPP,» [En línea]. Available: <https://www.ecured.cu/XAMPP>.
- [28] insomnia.rest, «Insomnia REST Client,» [En línea]. Available: <https://chrome.google.com/webstore/detail/insomnia-rest-client/gmodihnfibbjdecbanmpmbmeffmloel?hl=es-419>.
- [29] «¿Qué es MQTT?,» [En línea]. Available: <https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/>.
- [30] «Mosquitto,» [En línea]. Available: <https://aprendiendoarduino.wordpress.com/2018/11/19/mosquitto/>.
- [31] V. 3. SL, «AppLineup Dowload App,» [En línea]. Available: https://play.google.com/store/apps/details?id=com.veiem360.applineup&hl=es_419&gl=US.
- [32] «Asystel Volley Beach volleyball - voleibol Imágen de Png,» [En línea]. Available: <https://www.freepng.es/png-49h7wy/>.
- [33] P. y. m. -. Youtube, «Android desde Cero - Table Layout,» [En línea]. Available: https://www.youtube.com/watch?v=7bem0_wKINE.
- [34] StackOverflow, «Como crear un scroll view horizontal que se mueva automaticamente y ciclicamente,» [En línea]. Available: <https://es.stackoverflow.com/questions/79981/como-crear-un-scroll-view-horizontal-que-se-mueva-automaticamente-y-ciclicamente>.
- [35] R. F. E. d. Voleibol, «ActaOficialFIVB,» [En línea]. Available: <https://www.rfevb.com/RFEVB/Files/arbitros/documentacion/ActaOficialFIVB.pdf>.
- [36] J. R. D. -. Youtube, «Como leer PDF Viewer,» [En línea]. Available: <https://www.youtube.com/watch?v=QBOXFe9M03E>.
- [37] S. -. Youtube, «Make Android QR Code Generator App From Scratch | Android Studio Projects Tutorial For Beginners,» [En línea]. Available: <https://www.youtube.com/watch?v=NpVRUhEpRI8&t=158s-->.
- [38] J. T. Point, «Android Popup Menu Example,» [En línea]. Available: <https://www.javatpoint.com/android-popup-menu-example>.
- [39] N. C. -. Youtube, «spinner with radio button Android Studio,» [En línea]. Available: <https://www.youtube.com/watch?v=KGm-H0SHikY>.
- [40] A. C. -. Youtube, «How to Implement MultiSelect DropDown in Android Studio | MultiSelectDropDown | Android Coding,» [En línea]. Available: <https://www.youtube.com/watch?v=XrDVu3uPY3o>.
- [41] S. T. -. Youtube, «Create pdf in android using itext 7 | itext 7 android example | Android iText PDF example, iText PDF,» [En línea]. Available: <https://www.youtube.com/watch?v=Wntwt5XBv0k-->.

9. Anexo I – Manual de usuario

Este anexo contiene el manual de usuario donde se explica el funcionamiento de las diferentes pantallas de la aplicación Android con las que interactúa el usuario.

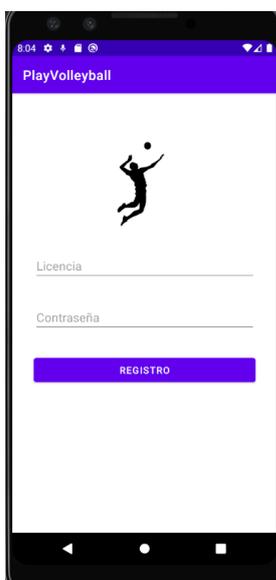
Login



Al entrar en la aplicación se accede al login donde se introducirá la licencia del entrenador o del árbitro con su respectiva contraseña. En caso de que no estén registrados, se muestra un mensaje de error. En caso de que el usuario no este registrado, se accede a registrar pulsando el botón registrar.

Ilustración 34. Imagen Login.

Registro



Se introduce la licencia y la contraseña deseada por el usuario que se almacenara en la base de datos. Si la contraseña es menos de 6 caracteres se mostrará un mensaje de error. Tras pulsar sobre el botón registro, se vuelve al login.

Ilustración 35. Imagen Registro.

Introducir rotación de los sets por parte de los entrenadores

En esta pantalla el entrenador introduce la rotación correspondiente al set y al lado donde se va a colocar.



Ilustración 36. Insertar rotación por parte del entrenador.

Información de los partidos

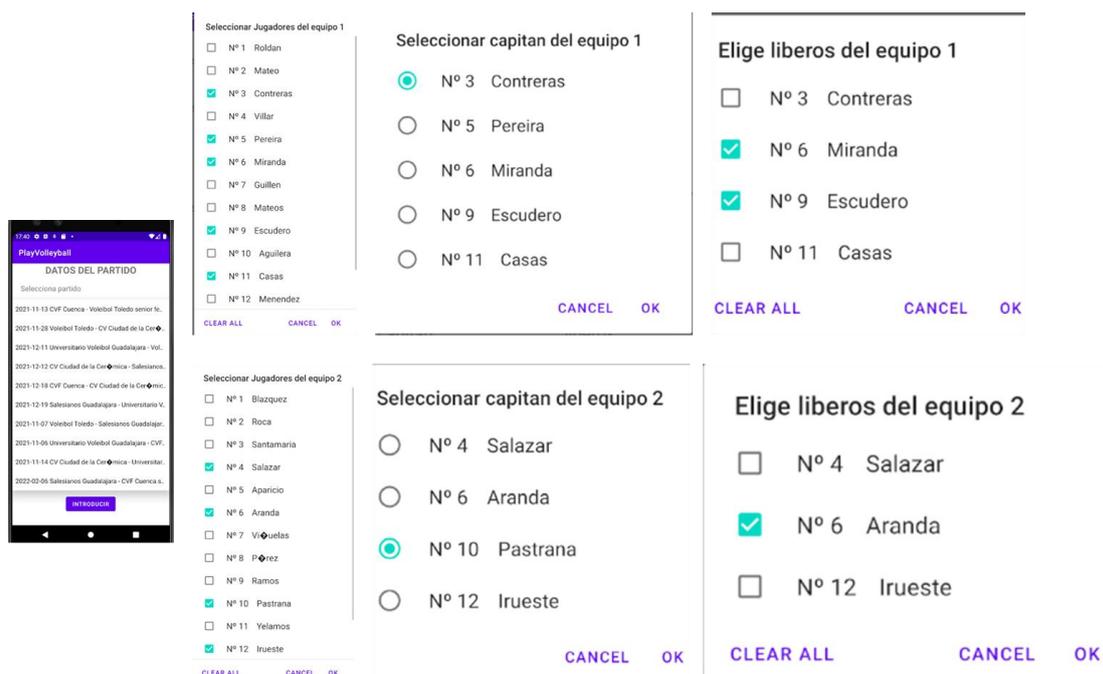


Ilustración 37. Imagen selección de datos.

A esta sección se accede si se hace login mediante la licencia de un árbitro. Aquí se puede seleccionar el partido que se va a jugar, los jugadores, capitanes y liberos de cada equipo.

Partido

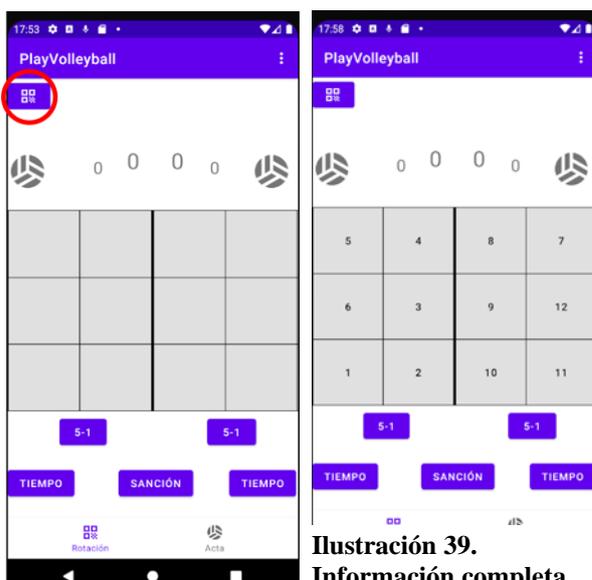


Ilustración 39.
Información completa
mediante código QR.

En esta sección se realiza toda la actividad principal del partido. Si se pulsa en el botón de la ilustración 35, se accede al lector de QR que inscribe la información en las posiciones correspondientes.

Ilustración 38. Botón QR.

A partir de aquí, se comienza a realizar el partido. Si se pulsa sobre la imagen del balón de voleibol, se establece el saque inicial. Después se debe pulsar sobre uno de los números del campo para establecer un punto o una sustitución. Si se establece un punto, el marcador subirá, si se realiza una sustitución, el jugador seleccionado cambiará por el numero introducido.

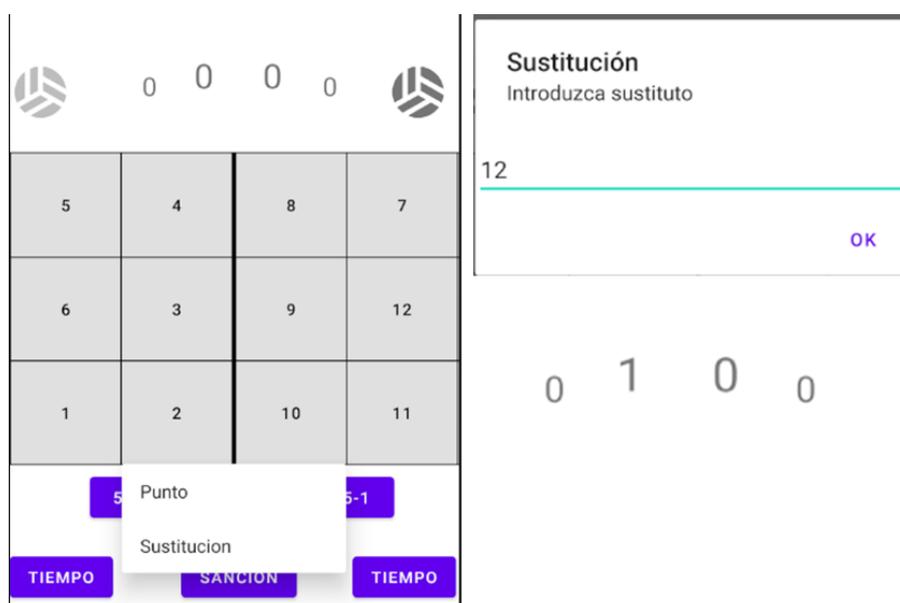
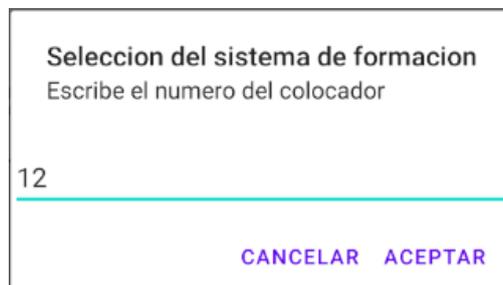


Ilustración 40. Establecer punto o sustitución.

En el caso de que se quiera ver el sistema de juego 5-1, se debe pulsar en uno de los botones correspondientes, dependiendo de que equipo se quiera ver el sistema. A partir de aquí, se introduce el número del colocador para ver la rotación pertinente.



Selección del sistema de formación
Escribe el número del colocador

12

CANCELAR ACEPTAR

Ilustración 41. Introducir el colocador en pista

También se puede solicitar hasta un máximo de dos tiempos por equipo pulsando el botón de tiempo de cada equipo. En caso de que se pulse más veces se muestra un mensaje.

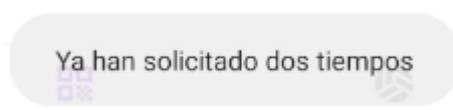
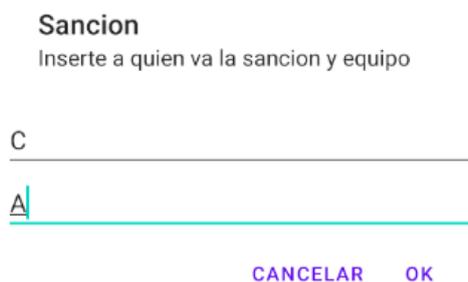


Ilustración 42. Mensaje de error al solicitar más de un tiempo.

Adicionalmente, se puede añadir sanciones, se debe introducir a quien va la sanción (C de Entrenador o el número del jugador) y a que equipo va dirigido (A o B).



Sancion
Inserte a quien va la sancion y equipo

C

A

CANCELAR OK

Ilustración 43. Insertar sanción.

Acta

Aquí se muestra toda la información obtenida a partir de los datos obtenidos en la pestaña anterior. A su vez, el acta se descarga en el almacenamiento local.

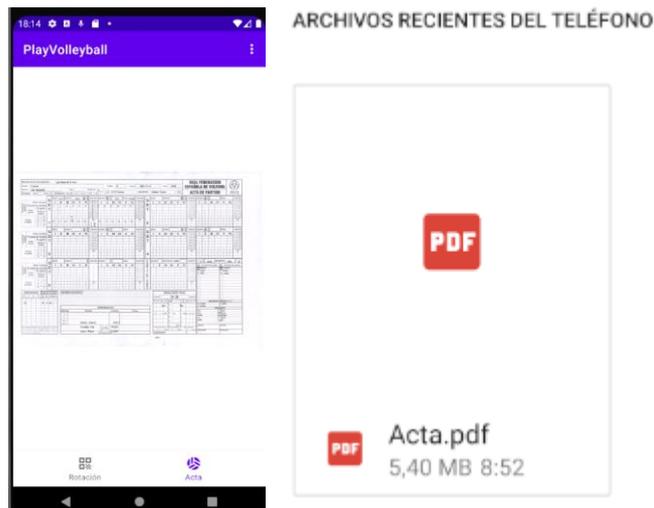


Ilustración 44. Acta descargada en el dispositivo.

Ilustración 45. Pantalla con el acta

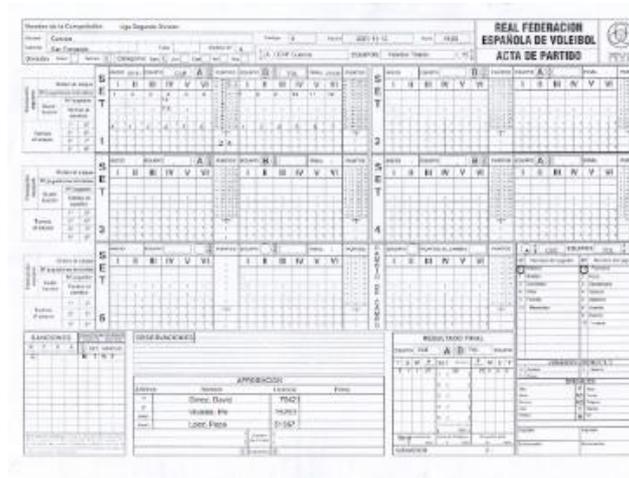


Ilustración 46. Acta con datos.

Menu tres puntos

En los tres puntos de la esquina superior derecha se puede acceder a la información de la aplicación o a cerrar la sesión.

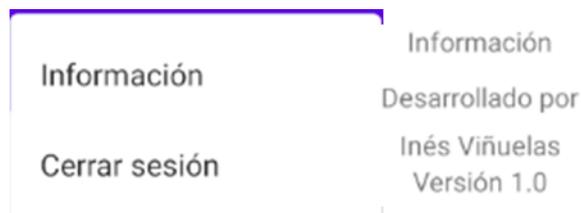


Ilustración 47. Imagen menú tres puntos.

10. Anexo II – Glosario

Punta o Receptor: Jugador balanceado con responsabilidad de ataque y recepción, su posición es de 4 en la zona de ataque y 5 en el zaguero.

Central: Jugador especializado en el bloqueo, su posición es de 3 en la zona de ataque y de 6 en el zaguero.

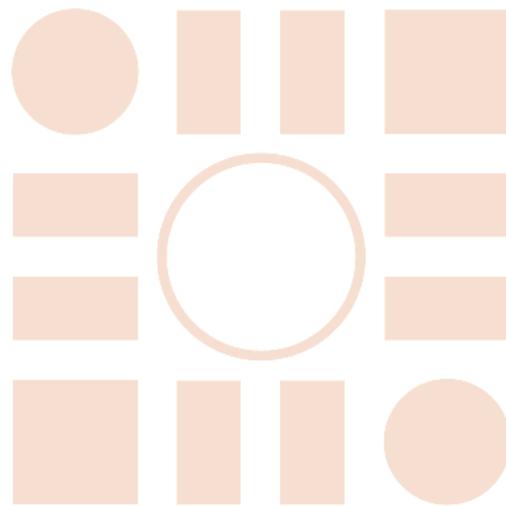
Líbero: Jugador puramente defensivo, entra por el central de la zona zaguero desde que pierde el saque y sale de la pista al rotar hacia 4 por lo que siempre estará en la zona defensiva, su posición es en 5.

Colocador: destinado a la repartir balón para que los atacantes realicen la acción de ataque. Su posición en juego es en la zona 2 de ataque y en la zona 1 en el área de zaguero.

Scout: Estadística de voleibol.

ScoutMan: Persona que realiza estadísticas de voleibol.

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá