Facultad 2

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Sistema para el análisis de acciones tácticas significativas de los equipos de balonmano

Autores: Ernesto Martínez Casanova

Dario Marcel Olavarrieta Martínez

Tutores: Ing. Vladimir Milián Núñez

Ing. Roberto Antonio Infante Milanés

La Habana, 2017

"...al final se sabrá que no es cuestión de títulos sino la utilidad de los saberes..."

Buena Fe

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los 16 días del mes de junio del año 2017.

Autores:

Ernesto Martínez Casanova	Dario Marcel Olavarrieta Martínez

Tutores:

Ing. Vladimir Milián Núñez	Ing. Roberto Antonio Infante Milanés

DEDICATORIA

“A mi madre”

Ernesto

“A mis padres”

Dario

AGRADECIMIENTOS

“A mis padres por haber hecho de mi la persona que soy, en especial a mi madre por todo el apoyo y por todo el sacrificio que hace día a día, por ser la mejor madre de este mundo.”

“A mis hermanos Joelito, Jorgito, Pedro y Lisi por su incondicional amistad.”

“A mi compañero de tesis Dario por ser un hermano para mí y por hacer que me divierta tanto haciendo la tesis con él.”

“A todos mis amistades de la universidad, a los que están y a los que no están, gracias por ser hermanos para mí.”

“A mis tutores Roberto y Vladimir por la ayuda brindada y por contribuir con mi formación como profesional.”

“A todo aquel que contribuyó con mi formación y que de una manera u otra me ha ayudado a lograr este sueño.”

A todos muchas gracias.

Ernesto

AGRADECIMIENTOS

“A mis padres, por darme todo el apoyo del mundo, que gracias a sus consejos y guías me han convertido en la persona que soy hoy.”

“A mis abuelos por ser mis segundos padres y por todo el cariño que me han dado.”

“A mis hermanos Claudia, Nayla y Noslen, por regalarme tantos momentos felices.”

“A mi tío Marcy por darme tantos consejos y haber despertado en mí el amor por la informática.”

“A mi novia Eileen, por el incondicional amor y apoyo que me ha brindado desde que nos conocimos, por hacerme parte de su vida y hacer más ameno estos 5 años.”

“A mi compañero de tesis Ernesto por tener tanta paciencia conmigo todo este tiempo, por compartir conmigo uno de los momentos más importantes de mi vida y lo más importante por ser mi amigo.”

“A mis tutores Roberto y Vladimir por toda la ayuda brindada.”

“Y a todos aquellos que de una forma u otra me apoyaron y ayudaron en todo este tiempo.”

Dario Marcel

RESUMEN

En el deporte una de las técnicas más usadas por los entrenadores para mejorar los resultados y el rendimiento del equipo es la recogida de la mayor cantidad de datos posibles durante el partido para su posterior análisis.

La presente investigación está basada en el desarrollo de un sistema para dispositivos móviles que utilicen Android como sistema operativo nombrado AndroHB, que permita el análisis de acciones tácticas significativas de los equipos de balonmano. Tiene como finalidad, obtener estadísticas claras y específicas de los partidos, jugadores, equipos y torneos, necesarias para enriquecer la información a tener en cuenta por los entrenadores, permitiéndoles tomar decisiones de formas más eficaces.

Asimismo, para su desarrollo se empleó la metodología AUP-UCI. Se utilizaron diversas tecnologías como el lenguaje de programación Java, el sistema gestor de base de datos *SQLite* y el entorno de desarrollo *Android Studio* para la implementación de las funcionalidades. Como resultado se una aplicación que agiliza la recogida de datos en un deporte tan dinámico como es el balonmano y es capaz de calcular de manera automática las estadísticas necesarias para facilitar el trabajo de los entrenadores.

Palabras claves: dispositivos móviles, Android, equipos de balonmano, estadísticas, toma de decisiones.

ÍNDICE GENERAL

ÍNDICE DE IMÁGENES

ÍNDICE DE TABLAS

INTRODUCCIÓN

Las Tecnologías de la Información y la Comunicación (TIC) constituyen un conjunto de instrumentos cada vez más eficaces en todos los aspectos de la sociedad. En la actualidad, entre las tecnologías más utilizadas en la gestión de la información están los dispositivos móviles inteligentes. Estos brindan notables ventajas como son la disminución de errores humanos en la captura de datos, el análisis en tiempo real, menos costos y más rendimiento, y la gran movilidad y simplicidad a la hora de la recogida de información.

Dentro de los sistemas operativos utilizados por este tipo de dispositivos se encuentran IOS, *Window Phone* y Android. De los antes mencionados el último fue creado por Google y es el único de código abierto. Entre sus principales características están que su núcleo se basa en el Kernel de Linux, es adaptable a muchas pantallas y resoluciones, utiliza SQLite para el almacenamiento de datos, soporta Java y muchos formatos multimedia, y cuenta con un catálogo de aplicaciones gratuitas o pagas llamado Google Play en el que pueden ser descargadas e instaladas. (1) Todas estas particularidades hacen que en la actualidad este sea el más utilizado de todos.

En la rama del deporte se evidencia la utilización de estos dispositivos móviles con sistema operativo Android debido a que existen aplicaciones utilizadas para llevar el control de la información de eventos deportivos de forma organizada, segura, accesible y rápida.

En el Balonmano moderno, los especialistas destacan las posibilidades que ofrece el control y la evaluación en el entrenamiento deportivo, la utilización de una planilla de evaluación-observación de medios tácticos, y la observación sistemática como técnica de medida para el análisis de la cuantificación y la evaluación táctica ofensiva, tanto individual como colectiva.

Se considera que se debe potenciar herramientas evaluativas propiciadoras de un espacio reflexivo, interactivo y motivador sobre las posibles acciones a desarrollar dentro del juego. Deben integrarse la técnica y la táctica en correspondencia con las exigencias del entrenamiento contemporáneo, siendo cada vez más similares a la forma de competición. Esto provoca un reajuste del proceso de control y evaluación en correspondencia con la forma de entrenamiento y de la competición, dada la identificación de los errores técnico-tácticos fundamentales en la estructura del movimiento en el balonmano de los equipos en competencia.

Con el fin de mejorar el rendimiento de los jugadores y el equipo, los entrenadores de balonmano desarrollan nuevas técnicas y tácticas de entrenamiento utilizando los

avances que ofrecen la ciencia y las tecnologías. Sin embargo, antes de formular cualquier nueva estrategia se necesita un basamento estadístico de cómo se están comportando los jugadores y el equipo en cada una de las fases del juego. Para lograr este basamento se debe realizar un control, análisis y evaluación de numerosas variables que corresponden a las acciones tácticas de un juego. Una acción táctica es un movimiento realizado por un jugador ya sea en ataque, defensa o contraataque. Algunas de las acciones tácticas más importantes que son recogidas durante los partidos son los goles, las asistencias, los fallos, las pérdidas de balón y las recuperaciones.

El proceso de recogida de dichas acciones en los torneos deportivos se torna en ocasiones engorroso, debido a las características tan dinámicas que posee el balonmano; además, llevar a cabo la recogida de los datos de forma manuscrita, hace que la durabilidad y organización de los mismos sea vulnerable, pues están más expuestos a perderse o a deteriorarse en un corto período de tiempo. Es por ello que la mayoría de los entrenadores han apostado por la utilización de los dispositivos móviles inteligentes como tecnología y medio para ayudar en el proceso de recogida de acciones tácticas en los torneos deportivos.

En pesquisas realizadas a los entrenadores de balonmano de la Habana con la colaboración del Jefe de la comisión de reglas y arbitraje del balonmano de la zona occidental de Cuba, Msc Lic Aloy Machado Sánchez, se obtuvo como resultado que de un total de 53 entrenadores, el 95% de los que poseen dispositivos móviles inteligentes (tabletas o celulares) utilizan como sistema operativo Android.

Teniendo en cuenta lo planteado anteriormente, se define como **problema a resolver:** ¿cómo facilitar a los entrenadores la recogida y análisis de acciones tácticas en los partidos de balonmano? Por lo que el **objetivo general** de este trabajo es: desarrollar un sistema de registro estadístico para el análisis del comportamiento de acciones tácticas significativas de los equipos de balonmano sobre el sistema operativo Android.

El **objeto de estudio** lo constituye el análisis estadístico aplicado en el deporte. En consecuencia el **campo de acción** está centrado en el proceso de gestión de información estadística en los partidos de balonmano.

Para dar cumplimiento al objetivo general se plantean las siguientes **tareas de investigación**:

1. Análisis de la bibliografía técnica del balonmano, para conocer las reglas del deporte y de las competiciones.

2. Análisis de sistemas existentes que permitan la recogida de resultados estadísticos en partidos de balonmano, para evaluar la factibilidad del desarrollo del sistema.
3. Análisis de las diferentes herramientas, tecnologías y metodología de desarrollo de software existentes, para conocer las ventajas que brindan y realizar la selección de las más convenientes.
4. Realización de entrevistas con los clientes que desean informatizar el proceso de recogida de acciones significativas en los juegos de balonmano, para identificar los requerimientos del sistema.
5. Desarrollo de la aplicación propuesta en correspondencia con las fases de la metodología seleccionada.
6. Ejecución de pruebas de software a las funcionalidades implementadas de la aplicación, para comprobar el correcto funcionamiento de la misma.

Para llevar a cabo el desarrollo de la investigación se utilizan los siguientes **métodos científicos:**

Métodos teóricos:

- Analítico-Sintético: este método permite analizar la documentación referente al balonmano para seleccionar los aspectos más importantes sobre el registro de acciones de juego en este deporte. Además, facilitará la selección de la metodología y las herramientas a utilizar para el desarrollo de la aplicación.
- Histórico-Lógico: para investigar sobre las aplicaciones similares existentes en Cuba y el resto del mundo.

Métodos empíricos:

- Entrevista (de forma no estructurada): para conocer cuáles son las necesidades de informatización de los profesionales del INDER y de la Federación Cubana de Balonmano con respecto a los partidos de balonmano.
- Modelación**:** utilizado para modelar la arquitectura, crear los artefactos, diagramas y modelos a utilizar en el desarrollo del Sistema para el análisis de acciones tácticas significativas de los equipos de balonmano.

El trabajo de diploma se divide en 3 capítulos, los cuales estarán estructurados de la siguiente forma:

Capítulo 1. Fundamentación teórica: se exponen los lenguajes, herramientas, y metodología a utilizar para el desarrollo de la aplicación. Además, se realiza un estudio

de otras soluciones existentes en el mundo vinculadas a facilitar el trabajo a los entrenadores de balonmano.

Capítulo 2. Propuesta de solución: se describe la propuesta de solución y se exponen los conceptos importantes del dominio a través del modelo conceptual. Se realiza la especificación de los requisitos funcionales y no funcionales y a partir de estos se definen los casos de uso del sistema. Además, se precisa la arquitectura y el patrón arquitectónico de la aplicación, así como los patrones de diseño utilizados y el modelo de datos.

Capítulo 3. Implementación y pruebas de la solución propuesta: se materializa la propuesta del sistema, describiendo el proceso con los diagramas de componentes, paquetes y de despliegue, y se valida el mismo a través de las pruebas de software.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se abordan elementos asociados al análisis de diferentes aplicaciones existentes, que brindan soporte a la toma de decisiones y la aplicación de estrategias por parte de los entrenadores de balonmano. Además, se hace un estudio de las principales metodologías de desarrollo de software, lenguajes, herramientas y tecnologías, así como la selección de las mismas en función del cumplimiento del objetivo general de la investigación.

1.1 Análisis de otras soluciones existentes

En el mundo existen aplicaciones informáticas que responden a las mismas necesidades que justifican el desarrollo de la presente investigación: gestionar el registro estadístico para el análisis del comportamiento de acciones tácticas significativas de los equipos de balonmano. Antes de comenzar a desarrollar el sistema propuesto se hizo necesario el estudio de esas aplicaciones para saber si es factible el desarrollo de una nueva aplicación o si sería mejor el uso de una de ellas. Para realizar esta valoración se tendrán en cuenta los siguientes aspectos: la gestión de jugadores, equipos y torneos; el cálculo automático de estadísticas; el código abierto; el registro en tiempo real de acciones; la pizarra táctica; y que trabaje sobre el sistema operativo android.

1.1.1 Aplicaciones internacionales

Handball Manager 12

Aplicación implementada por el grupo de desarrolladores MantuApps para dispositivos con sistema operativo Android, ofrece una gran oportunidad de planificar y mostrar las tácticas de juego del equipo. Puede colocar jugadores y pelotas para el campo y fácilmente numerar los jugadores, moverlos en el campo y dibujar líneas para mostrar a su equipo cómo jugar. (3)

El *Handball Manager* cuenta con las siguientes características y funcionalidades:

- Salvar y cargar tácticas
- Salvar y cargar colores
- Añadir y eliminar jugadores
- Gráficos de alta calidad
- Menús fáciles de usar
- Números de los jugadores

- Lápiz de dibujo
- Editor de colores
- Instrucciones en la aplicación.

Como se puede apreciar este sistema tiene elementos que pueden ser importantes para la construcción del software que se quiere lograr mediante este trabajo, como por ejemplo la oportunidad de mostrar las tácticas de juego en el tablero utilizando el lápiz de dibujo, así como la gestión de los jugadores incluyendo su número de camiseta y la gama de colores con que cuenta. También, es importante mantener la interfaz del menú sencilla y fácil de usar. Es una aplicación muy útil para los entrenadores de balonmano a la hora de dar instrucciones al equipo, pero se limita bastante a lo que se quiere lograr con este trabajo porque no permite la gestión de equipos y torneos, no lleva el conteo del marcador durante un partido, no permite el cálculo de estadísticas de los jugadores en los partidos, es necesario pagar una licencia para utilizarla y no es de código abierto.

Assistant Coach Handball

Aplicación completa y profesional para entrenadores de balonmano desarrollada por Valerio Lo Giudice para gestionar equipos, jugadores, partidos, estadísticas, entrenamientos, ejercicios y cuerpo técnico. Trabaja sobre el sistema operativo Android. Permite agregar equipos por temporada y ver la información de cada uno y las estadísticas de la temporada; registrar jugadores con todos sus datos y foto incluida, así como calcular automáticamente las estadísticas de estos; añadir planes de entrenamiento y ejercicios; y agregar resultados de los partidos y el rendimiento de cada jugador después del partido. También, permite exportar todos los informes estadísticos y los planes de entrenamiento en PDF (*Portable Document Format*) y enviarlos por correo electrónico o importarlos al ordenador. (4)

La aplicación cuenta con las siguientes características:

- Gestionar información de entrenamientos, jugadores, equipos y partidos de cada temporada.
- Añadir foto del equipo y de los jugadores.
- Exportación en formato PDF de estadísticas de equipo, de juego y de jugadores; planes de entrenamiento y los perfiles de los jugadores.
- Agregar un nuevo entrenamiento con descripción y la fecha.
- Grabar un video de tus jugadores realizando cada ejercicio del entrenamiento.

- Dibujar el ejercicio de entrenamiento en la pizarra.
- Añadir el rendimiento individual de cada jugador después de un partido y la aplicación calculará las estadísticas individuales del jugador y las totales del equipo.

En general esta es una aplicación muy completa y útil para los entrenadores de balonmano. De ella se pueden tomar numerosas ideas para el software que se quiere desarrollar, como es el caso de las múltiples opciones para gestionar los jugadores, partidos, equipos y entrenamientos, destacando la gran cantidad de información que recoge de cada categoría incluyendo fotos de jugadores y equipos. Además, realiza el cálculo automático de estadísticas al final de cada partido y cada temporada, funcionalidades necesarias en la aplicación que se quiere desarrollar. Por otro lado tiene como limitación que no es un software gratuito, no es de código abierto y no permite llevar el control en tiempo real de un partido, anotando las jugadas, tácticas y estrategias específicas de los equipos durante el mismo, restringiendo de esta forma algunas estadísticas importantes.

Estadísticas Balonmano

Aplicación Android que permite crear estadísticas profesionales de todos los partidos de equipos de balonmano, acceder a ellas en tiempo real durante el partido o como evaluación de toda la temporada. Ideal para que entrenadores y gestores identifiquen fortalezas y debilidades de cada jugador y equipo y mejoren sus habilidades. Debido a la rapidez de los partidos, esta aplicación presta especial atención a una introducción de datos rápida e intuitiva, cada acción puede introducirse en dos toques, escogiendo de listas concisas. Permite añadir equipos, jugadores y varios tipos de torneos. En preferencias es posible elegir las acciones que tienen que introducirse y considerarse en las estadísticas y el usuario puede definir hasta 10 acciones. Brinda estadísticas completas para cada partido disponibles durante el mismo. Para una vista completa de toda la temporada permite generar estadísticas del torneo que contienen a todos los jugadores que jugaron al menos un partido, el número de partidos, las acciones y los tiros y niveles de consolidación. (5)

De las estadísticas que brinda cabe destacar:

- estadísticas individuales que contienen cada acción de los jugadores del partido actual, incluidos los tiros y los niveles de consolidación.
- las estadísticas del partido con un resumen de todas las acciones desglosadas en primer y segundo tiempo y una vista general de la distribución de los goles.

- la imagen completa del juego, con cada acción, el tiempo y el resultado en ese momento.
- todas estas estadísticas pueden enviarse por correo electrónico.

Esta es una aplicación muy completa y profesional, que les permite a los entrenadores llevar una detallada información sobre un partido de balonmano. En base a la aplicación que se quiere desarrollar se pueden tomar muy buenas ideas de esta, tales como la gestión en tiempo real de las acciones durante el partido, la gran gama de estadísticas que brinda, la interfaz de usuario, la gestión de jugadores, equipos, partidos y torneos, y la opción que brinda para definir 10 nuevas acciones de juego que se tienen en cuenta en las estadísticas. No obstante tiene también sus limitaciones con respecto a lo que se quiere lograr, no cuenta con una pizarra que permita dibujar acciones tácticas, no es gratuita por lo que se hace más complicada su adquisición, además, no permite acceder a su código fuente por lo que imposibilita la reutilización del mismo.

Handball Board

Aplicación Android desarrollada por TOPROOM Shinagawa que consiste en una pizarra de balonmano, diseñada para que el entrenador de un equipo pueda explicar las tácticas y jugadas a los jugadores, además, permite guardar y cargar datos de los jugadores en la ventana de configuración. También, brinda la opción de reproducir jugadas que hayan sido guardadas por el usuario. (6)

Principales características:

- Cambiar el tamaño del campo (completo o medio).
- Guardar datos de jugadas.
- Cargar datos de jugadas.
- Reproducción de jugadas.
- Cambiar el número y el nombre de los jugadores.

Esta es una aplicación bastante sencilla en comparación con lo que se quiere desarrollar, se pueden aprovechar las ideas de la pizarra a la hora de guardar jugadas y reproducirlas, así como la opción de cambiar el tamaño del campo y la combinación de colores que se utiliza. La aplicación tiene gran cantidad de limitaciones ya que no brinda ningún tipo de estadísticas ni permite gestionar información, solo representar jugadas en la pizarra, además, se necesita pagar una licencia para su adquisición.

Estadísticas *Handball*

Pequeña aplicación Android pensada para llevar las estadísticas de todo un equipo de balonmano mientras juega. Tiene opciones para adicionar jugadores y partidos, ver la composición del equipo, las estadísticas individuales de los jugadores y las del equipo durante el partido. La interfaz es sencilla y fácil de utilizar. (7)

Los elementos más importantes que se pueden aprovechar de esta para la construcción del software que se quiere lograr mediante este trabajo son la sencillez de la interfaz, las opciones de registro de jugadores y partidos, y las estadísticas que brinda. Tiene varias limitaciones importantes como la gestión de torneos, de varios equipos, de los entrenamientos; además, es una aplicación de pago, brinda pocas estadísticas y no cuenta con una pizarra para la explicación de las jugadas.

Scoreboard Handball

Aplicación Android basada en un marcador de balonmano que muestra el nombre de los equipos, el puntaje, el tiempo y el período del partido. Tiene opciones para modificar los minutos por períodos y el número de períodos regulares del partido, y para habilitar y deshabilitar el tiempo de exclusión y utilizarlo en el marcador. (8)

Es una aplicación muy sencilla de la cual se pueden tomar ideas para la interfaz del marcador de la aplicación a desarrollar y las opciones de modificación antes mencionadas. Sin embargo, se limita a llevar el marcador del partido, por lo que no cumple las necesidades del cliente.

1.1.2 Aplicaciones nacionales

Sistema de Anotación del Balonmano

El Sistema de Anotación del Balonmano permite llevar el control de un conjunto de datos que se recogen durante los partidos de este deporte con el objetivo de ser analizados y que permitan tomar decisiones. Permite llevar el control de un partido en tiempo real, brindando al anotador unas tablas en las cuáles podrá recoger los datos predeterminados en ellas de forma rápida y sencilla teniendo apoyo del mouse y del teclado, además, permite la gestión de partidos, campeonatos, equipos y jugadores. (9)

Brinda opciones para:

- Consultar Datos de un Juego.
- Consultar Datos de un Equipo en un Campeonato.
- Crear Campeonato.
- Insertar Equipos.

- Iniciar juego.

Es una aplicación que brinda una gran cantidad de datos para facilitar la toma de decisiones a los entrenadores de los equipos de balonmano. Cuenta con una serie de características muy útiles que encajarían en la aplicación a desarrollar, se pueden tomar las ideas de la gestión de los campeonatos y de los equipos que participan en estos, también la forma que presenta para la recogida de información en tiempo real de los partidos y cuáles son los datos que recoge. Como limitaciones, la principal desventaja de esta aplicación es que no corre sobre el sistema operativo Android, por lo que no puede ser utilizada en dispositivos móviles y tabletas, restándole sencillez y velocidad al proceso de recogida de datos durante los partidos. Además, no recoge todas las estadísticas que se quieren en la aplicación a desarrollar y carece de funcionalidades para ayudar a los entrenadores a diseñar los planes de entrenamiento de los jugadores.

1.1.3 Conclusiones del estudio de las herramientas

Después de realizado el estudio de las soluciones existentes a nivel nacional e internacional, se concluyó que la mayoría son soluciones propietarias con licencia de pago que no permiten acceder a su código fuente imposibilitando que este pueda ser reutilizado, y que ninguna brinda la totalidad de funcionalidades o de estadísticas que se necesita. Por estas razones no se elige ninguna de estas aplicaciones como solución al problema planteado y se decide desarrollar la aplicación AndroHB que contará con las funcionalidades necesarias para satisfacer todas las necesidades del cliente.

1.2 Metodología de desarrollo de software

Una metodología impone un proceso de forma disciplinada sobre el desarrollo de software con el objetivo de hacerlo más predecible y eficiente. Una metodología define una representación que permite facilitar la manipulación de modelos, y la comunicación e intercambio de información entre todas las partes involucradas en la construcción de un sistema. (10)

Para el desarrollo de software, se requiere de diversos elementos que agrupados hacen que este proceso sea o no exitoso. Para esto existen las metodologías tradicionales que se modificaron para poder aplicarlas al desarrollo de software, aunque durante mucho tiempo fueron la única solución, luego de varios años se hicieron poco flexibles y muy cuadriculadas. Estas consistían en una serie fundamentos y conceptos aplicados al desarrollo de software, documentación, planificación y procesos. (11)

Ante las dificultades de las metodologías tradicionales referentes al tiempo y flexibilidad, aparecen las metodologías ágiles como una respuesta metodológica, especialmente porque están orientadas a proyectos pequeños, constituyen una solución a la medida del entorno, simplificando las prácticas y asegurando la calidad del producto. (11)

En búsquedas realizadas para determinar un método que pudiera ser usado en la determinación del enfoque y la metodología para la ejecución del proyecto, se encontraron métodos como el Gráfico propuesto por Boehm y Turner, Método de Expertos y la Matriz de Evaluación de Metodología. Se selecciona el modelo propuesto por Boehm y Turner debido a que es un modelo eficiente, de fácil comprensión y uso, capaz de evaluar, cuantificar e identificar cinco variables críticas a la hora de decidir si el desarrollo de un sistema se efectúa por metodologías ágiles o robustas.

Este método plantea 5 criterios fundamentales mediante los que se estará valorando el proyecto; estos son: tamaño del equipo, criticidad del producto, dinamismo de los cambios, cultura del equipo y personal con que se cuenta. Para la selección del valor que se ubicará en cada eje (uno para cada criterio) de la estrella se debe tener en cuenta el comportamiento de estos criterios en el proyecto. (12)

En lo sucesivo se describe cada uno: (12)

- Tamaño: este criterio se utiliza para representar el número de personas involucradas en el proyecto. Pueden tenerse en cuenta el nivel de complejidad que pueda presentarse en la comunicación entre los miembros del proyecto y los costos que pueden provocar cambios esperados.
- Criticidad: se utiliza para evaluar la naturaleza del daño ocasionado por defectos que no hayan sido detectados al producto. Su evaluación puede ser cualitativa.
- Dinamismo: representa la rapidez con la que pueden estar cambiando los requerimientos del proyecto.
- Personal: representa la proporción del personal con experiencia alta, media y baja.
- Cultura: las organizaciones y las personas que relaciona el proyecto pueden depender de la confianza o de la relación contractual. Esto refleja el nivel de ceremonia necesario y aceptado: documentación, control, formalismo en las comunicaciones.

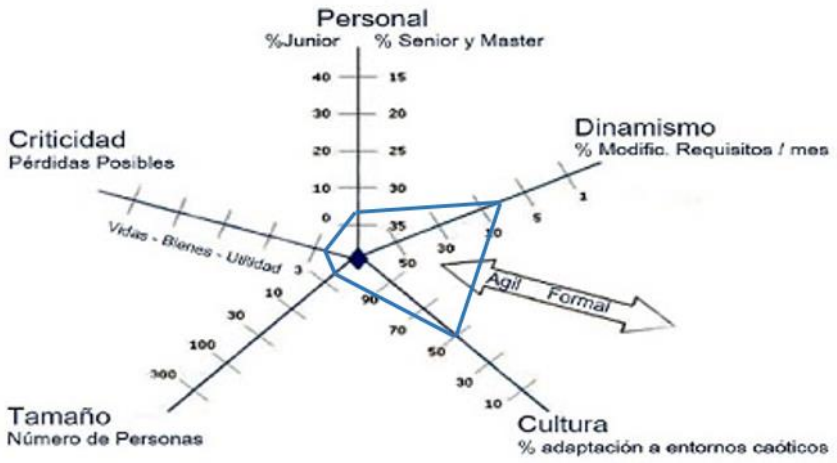


Figura 1: Aplicación del método de Boehm y Turner

Analizando cada uno de estos 5 criterios fundamentales en correspondencia con las características del equipo, se llega a la conclusión que el proyecto debería utilizar una metodología ágil.

1.2.1 Selección de la metodología

Programación Extrema (*eXtreme Programming*, XP por sus siglas en inglés) es una metodología ágil que se centra en potenciar las relaciones interpersonales como clave para el éxito, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. (13)

XP se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. (13)

Emplea historias de usuario para describir las funcionalidades del sistema que luego serán desarrolladas mediante la programación en parejas, permitiendo que los programadores menos experimentados aprendan de los más experimentados. Propone además, planificar un máximo de 40 horas semanales de trabajo evitando el sobreesfuerzo del equipo de desarrollo.

El ciclo de desarrollo de XP es el siguiente:

1. El cliente define el valor de negocio a implementar
2. El programador estima el esfuerzo necesario para su implementación
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo
4. El programador construye ese valor de negocio

5. Vuelve al paso 1

Proceso Unificado Ágil de Scott Ambler o *Agile Unified Process* (AUP) en inglés es una versión simplificada del Proceso Unificado de Desarrollo (*Rational Unified Process*, RUP por sus siglas en inglés). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo: (14)

- Desarrollo dirigido por pruebas.
- Modelado ágil.
- Gestión de cambios ágil.
- Refactorización de base de datos para mejorar la productividad.

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva: (14)

1. Inicio: el objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.
2. Elaboración: el objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
3. Construcción: durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.
4. Transición: el sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.

Para el desarrollo de la aplicación se seleccionó como metodología de desarrollo AUP en una variante desarrollada en la Universidad de las Ciencias Informáticas (UCI), nombrada AUP-UCI. Esta metodología fue ideada para tomar aspectos importantes de varias metodologías ágiles con las que se trabaja en la universidad y unificarlos en esta debido a que en ocasiones los proyectos no convergen en una sola metodología de desarrollo según lo descrito por Sánchez en 2015.

AUP-UCI fue seleccionada debido a que se centra en equipos de desarrollo pequeños, como este caso que está integrado solamente por dos programadores de mediana experiencia. Además, brinda una gestión de cambios ágil que hace que el equipo de desarrollo se adapte a nuevas condiciones, que en este caso los requisitos pueden

cambiar a decisión del cliente. Otro punto a su favor es la forma en que se planifica el proyecto y realiza la estimación del tiempo, pues se cuenta con poco tiempo para el desarrollo de la aplicación. Finalmente como factor más determinante de esta selección es que esta es la metodología utilizada actualmente en la UCI.

En AUP-UCI a partir de que el modelado de negocio propone tres variantes a utilizar en los proyectos (Casos de Uso del Negocio, Descripción de Procesos de Negocio, Modelo Conceptual) y existen tres formas de encapsular los requisitos (Casos de Uso del Sistema, Historias de Usuario, Descripción de Requisitos por Proceso), surgen cuatro escenarios para modelar el sistema en los proyectos. (14)

Para el desarrollo de este trabajo se seleccionó el escenario No 2, pues una vez evaluado el negocio a informatizar se determinó que no es necesario incluir las responsabilidades de las personas que van a ejecutar las actividades, por tanto solo será necesario modelar exclusivamente los conceptos fundamentales del negocio. Además, en el presente proyecto el objetivo primario es la gestión y presentación de información, que es precisamente lo que plantea este escenario. (14)

1.3 Lenguajes, herramientas y tecnologías

Para llevar a cabo el desarrollo de la aplicación es necesario la utilización de varias herramientas. A continuación se describen los lenguajes, herramientas y tecnologías empleadas para la modelación, diseño e implementación de la aplicación.

1.3.1 Lenguajes

Lenguaje de programación

Java es un lenguaje de programación orientado a objetos que se popularizó a partir del lanzamiento de su primera versión comercial de amplia difusión, la JDK 1.0 en 1996. Se utiliza en su versión 1.8.0_77. (15)

Además de ser orientado a objeto, otras de las características que lo hacen ser uno de los lenguajes más usados para la programación en todo el mundo, son las siguientes: (15)

- Simple
- Distribuido
- Interpretado
- Sólido
- Seguro

- Arquitectura neutral
- Portable

Lenguaje de modelado

UML son las siglas de *Unified Modeling Language* o Lenguaje Unificado de Modelado. Es un estándar internacional adoptado por numerosos organismos y empresas que plantea una serie de normas gráficas que indican la forma de representar los esquemas, diagramas y documentación relativa al desarrollo de un software. (16)

De los variados elementos que brinda UML para representar las diferentes partes de un sistema, se utilizan en el desarrollo de la aplicación los diagramas de casos de uso, de despliegue, de componentes, de colaboración y el resto de los diagramas. Se utiliza en su versión 2.5.

1.3.2 Herramientas y tecnologías

Entorno de desarrollo integrado

Android Studio es el entorno de desarrollo integrado (*Integrated Development Environment*, IDE por sus siglas en inglés) oficial de Android. Está diseñado para que Android pueda acelerar el desarrollo y permita crear las aplicaciones de mejor calidad para todos los dispositivos de que utilizan este sistema operativo. Ofrece herramientas personalizadas para programadores de Android. Se incluyen herramientas completas de edición, depuración, pruebas y perfilamiento de códigos. Se utiliza en su versión 2.1.3.

Algunas de sus características son: (17)

- Emulador rápido y cargado de funciones: *Android Emulator* se instala e inicia las aplicaciones más rápido que un dispositivo real, también permite crear prototipos de estas y probarlas en todas las configuraciones de dispositivos Android: teléfonos, tablets y dispositivos *Android Wear* y *Android TV*. Además, permite simular varias funciones de hardware, como la localización de GPS, la latencia de red y las funciones multitáctiles.
- Editor de código inteligente: al ofrecer compleción avanzada de código, refactorización y análisis de código, el editor de código inteligente permite escribir un código más eficaz, trabajar más rápido y ser más productivo. A medida que se escribe, *Android Studio* proporciona sugerencias en una lista desplegable. Simplemente presiona Tab para insertar el código.

- Sistema de compilación sólido y flexible: ofrece automatización de compilaciones, administración de dependencias y configuraciones de compilación personalizables. Permite configurar un proyecto de modo que se incorporen bibliotecas locales y alojadas, y definir variantes que incluyan código y recursos diferentes, además de aplicar configuraciones de reducción de código y firma de aplicaciones.

Sistema gestor de bases de datos

SQLite es una biblioteca que implementa un motor de bases de datos de Lenguaje de Consulta Estructurada (*Structured Query Language* o SQL por sus siglas en inglés) autónomo, sin servidor, sin configuración y transaccional. El código para SQLite está en el dominio público y por lo tanto es gratuito para cualquier propósito, comercial o privado. SQLite es la base de datos más implementada en el mundo con más aplicaciones de las que podemos contar, incluyendo varios proyectos de alto perfil. (18)

Sus principales características son: (18)

- Es un sistema completo de bases de datos que soporta múltiples tablas, índices, triggers y vistas. No necesita un proceso separado funcionando como servidor ya que lee y escribe directamente sobre archivos que se encuentran en el disco duro. El formato de la base de datos es multiplataforma e indistintamente se puede utilizar el mismo archivo en sistemas de 32 y 64 bits.
- La base de datos se almacena en un único fichero a diferencia de otros sistemas gestores de bases de datos que hacen uso de varios archivos. SQLite emplea registros de tamaño variable de forma tal que se utiliza el espacio en disco que es realmente necesario en cada momento.
- El código fuente está pensado para que sea entendido y accesible por programadores promedio. Todas las funciones y estructuras están bien documentadas.

Para el desarrollo de la solución propuesta se utiliza la librería de Java ***Object Relational Mapping Lite*** (*ORMLite*) que ofrece algunas funciones ligeras para la persistencia de objetos Java a bases de datos SQL evitando al mismo tiempo la complejidad y los gastos generales de más paquetes ORM estándar. Es compatible con una serie de bases de datos SQL mediante *Java Database Connectivity* (JDBC), con SQLite y con las llamadas nativas a las Interfaces de Programación de Aplicaciones (API por sus siglas en inglés) de base de datos del sistema operativo Android. (19)

Algunas de sus características son: (19)

- Agilidad en el desarrollo software.
- Son usados de manera mayoritaria en sistemas de Bases de Datos SQL.
- Mantenimiento y reutilización de código.
- Encapsulación de datos: se mantiene la integridad de los datos ocultando su estado o la de los datos miembros de un objeto, de forma que sólo se permite su modificación a través de los métodos definidos para dicho objeto.

Herramienta de modelado UML

Se utiliza ***Visual Paradigm*** 8.0, herramienta para desarrollo de aplicaciones utilizando modelado UML, es ideal para ingenieros de software, analistas y arquitectos que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos. (20)

Algunas de las características que ofrece esta herramienta son: (20)

- Navegación intuitiva entre la escritura del código y su visualización.
- Ambiente visualmente superior de modelado.
- Potente generador de informes en formato PDF/HTML.
- Sofisticado diagramador de *layout*.

Conclusiones del capítulo

En este capítulo se analizaron algunas aplicaciones existentes para apoyar a los entrenadores de balonmano en la toma decisiones y la dirección de los partidos. Para este análisis se tuvo en cuenta fundamentalmente la recogida de estadísticas, permitiendo distinguir las ventajas que trae consigo el desarrollo de la aplicación propuesta en el presente trabajo. Se investigó sobre las metodologías de desarrollo de software más utilizadas para guiar el desarrollo de este tipo de aplicaciones, seleccionándose AUP-UCI. Además, se definieron las herramientas y tecnologías a emplear teniendo en cuenta las características de estas y las necesidades para el desarrollo de la propuesta de solución. Se seleccionó Java 1.8.0_77 como lenguaje de programación, *Visual Paradigm* 8.0 para el modelado UML, *Android Studio* en su versión 2.1.3 como IDE y SQLite como sistema gestor de bases de datos, con la librería *ORMLite* para el mapeo de los datos.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

En el presente capítulo se describe la solución propuesta utilizando los artefactos que propone la metodología AUP-UCI. Se presenta el modelo de dominio donde se analizan las entidades y conceptos relacionados con el desarrollo de AndroHB. Se definen los requisitos funcionales y no funcionales del sistema a desarrollar, así como el diagrama de casos de uso del sistema.

2.1 Propuesta de solución

Para dar solución al problema tratado en esta investigación, se propone el desarrollo de la aplicación AndroHB. Su objetivo es facilitar el trabajo a los entrenadores de balonmano, sirviendo de apoyo en la toma de decisiones y agilizando el proceso de recogida de datos y cálculo de numerosas estadísticas.

La solución propuesta permitirá gestionar torneos, equipos, jugadores y partidos de balonmano. La gestión de los partidos se hará en tiempo real, permitiendo la anotación de las principales acciones durante el juego de una forma rápida y sencilla, y partiendo de estas la aplicación calculará y mostrará automáticamente una serie de estadísticas, brindándole al entrenador una información importante para la toma de decisiones. Será posible acceder a las estadísticas tanto durante el partido como después de finalizado a través del menú de la aplicación, donde se brindará además información de los torneos, equipos y jugadores.

Figura 2: Diagrama de propuesta de solución

2.2 Modelo conceptual

El modelo conceptual es una representación visual de clases conceptuales o de objetos reales en el dominio del problema. No constituyen clases de software, son representaciones de los conceptos involucrados para la comprensión de la solución y las asociaciones establecidas describen las relaciones entre los mismos. (21)

La figura 3 muestra los conceptos importantes del dominio de la solución a desarrollar mediante el modelo conceptual.

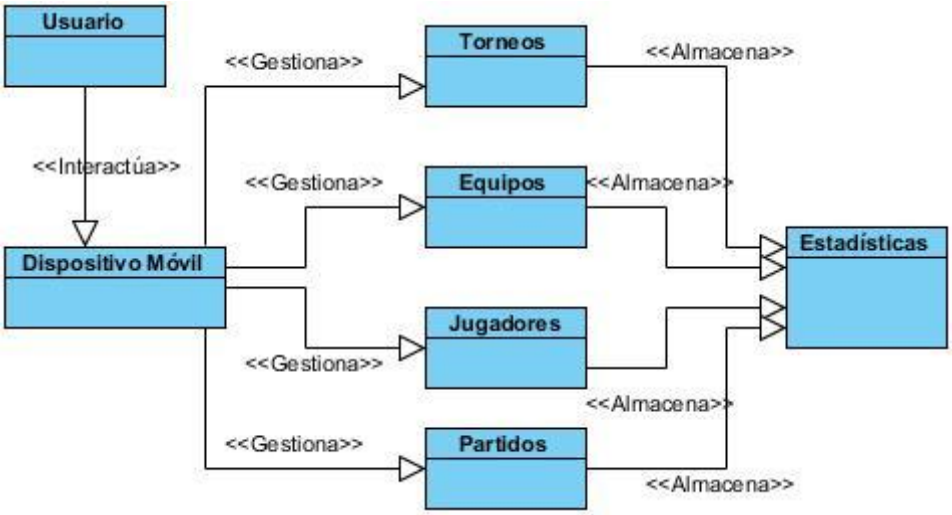


Figura 3: Modelo Conceptual

A continuación se presenta una descripción de los conceptos involucrados en el modelo conceptual:

Usuario: es la persona que está interactuando con la aplicación, dígase un miembro del cuerpo técnico del equipo de balonmano, el entrenador o un árbitro.

Dispositivo Móvil: es la entidad que hace referencia al medio donde se encuentra instalada la solución propuesta.

Torneo: hace referencia al evento principal de balonmano, donde intervienen jugadores, equipos y partidos.

Equipo: es el colectivo integrado por varios jugadores.

Jugador: es la persona que juega en un partido de balonmano.

Partido: forma de competición donde se enfrentan dos equipos.

Estadísticas: hace referencia a todas las estadísticas que se calculan tanto de los partidos, como de los equipos, jugadores y torneos. Las estadísticas se calculan a partir de las acciones significativas anotadas por el usuario.

2.3 Requisitos funcionales del sistema

Tabla 1: Listado de requisitos funcionales

Número del requisito	Nombre del requisito	Descripción
RF1	Gestionar torneos	Permite al usuario

		gestionar la información referente a los torneos.
RF1.1	Insertar torneo.	Permite al usuario añadir a la base de datos un nuevo torneo.
RF1.2	Detalles torneo	Muestra los resultados de los partidos jugados en el torneo.
RF1.3	Listar torneos.	Lista todos los torneos existentes en la base de datos y los muestra al usuario
RF1.4	Modificar torneo	Permite al usuario modificar un torneo seleccionado de la lista.
RF1.5	Eliminar torneo.	Permite al usuario eliminar un torneo.
RF2	**Gestionar jugadores**	**Permite al usuario gestionar la información referente a los jugadores.**
RF2.1	Insertar jugador.	Permite al usuario añadir a la base de datos un nuevo jugador.
RF2.2	Listar jugadores.	Lista todos los jugadores existentes en la base de datos.
RF2.3	Modificar jugador	Permite al usuario modificar un jugador seleccionado de la lista.

RF2.4	Eliminar jugador.	Permite al usuario eliminar un jugador.
RF2.5	Detalles jugador	Muestra la información referente a un jugador seleccionado de la lista.
RF2.6	Exportar estadísticas a PDF	Exporta las estadísticas de un jugador a un archivo con extensión PDF
RF3	**Gestionar equipos.**	**Permite al usuario gestionar la información referente a los equipos.**
RF3.1	Insertar equipo.	Permite al usuario añadir a la base de datos un nuevo equipo.
RF3.2	Listar equipos.	Lista todos los equipos existentes en la base de datos.
RF3.3	Modificar equipo	Permite al usuario modificar un equipo seleccionado de la lista.
RF3.4	Eliminar equipo.	Permite al usuario eliminar un equipo.
RF3.5	Detalles equipo	Muestra al usuario la información referente a un equipo.
RF3.6	Exportar estadísticas a PDF	Exporta las estadísticas de un equipo a un archivo con extensión PDF
RF4	**Crear nuevo partido**	**Permite al usuario crear**

		un nuevo partido a partir de un torneo y equipo previamente creados.
RF4.1	Cronómetro de juego	Genera un cronómetro que lleva el control del tiempo del partido.
RF4.2	Cronómetros de amonestaciones	Genera un cronómetro cada vez que un jugador es amonestado y expulsado del partido por un tiempo.
RF4.3	Insertar acción	Permite al usuario insertar una acción significativa del partido.
RF4.4	Pizarra táctica	Muestra una pizarra táctica que permite al entrenador dar indicaciones al equipo.
RF4.5	Calcular estadísticas	Calcula automáticamente las estadísticas del partido en tiempo real a partir de las acciones insertadas.

2.4 Requisitos no funcionales del sistema

La propuesta de solución debe presentar un conjunto de características que ayuden a mejorar el rendimiento, confiabilidad y usabilidad de la misma. Para ello se necesitan establecer varios requisitos no funcionales.

Usabilidad:

- La aplicación debe tener una interfaz intuitiva, de manera que permita al usuario sin experiencia interactuar fácilmente con el sistema.
- El sistema se mostrará solo de manera vertical.

Apariencia e Interfaz:

- Todos los textos y mensajes en pantalla aparecerán en idioma español.
- La gama de colores a utilizar será blanco, azul y negro.

Hardware:

- Será necesario disponer de un dispositivo móvil inteligente, puede ser un teléfono celular o una tableta.
- El dispositivo debe tener como mínimo: un procesador de Intel Atom® Z2520 a 1,2 GHz, 512Mb de memoria RAM y 80Mb de espacio disponible.

Software:

- El dispositivo móvil debe tener un sistema operativo Android versión 4.0 o superior.
- Compatibilidad con ORMLite en su versión 4.48.

2.5 Definición de los casos de uso

A partir del análisis de los requisitos funcionales del sistema se definieron los siguientes casos de uso:

- CU1: Gestionar torneos
- CU2: Gestionar equipos
- CU3: Gestionar jugadores
- CU4: Crear nuevo partido

2.5.1 Actores del sistema.

Actor	Descripción
Usuario	Es la persona que interactúa con la aplicación, dígase un miembro del cuerpo técnico del equipo de balonmano, el entrenador o un árbitro. Será el encargado de introducir la información necesaria sobre los torneos, equipos, jugadores y partidos.

2.5.2 Diagrama de casos de uso del sistema

Un diagrama de Casos de Uso del Sistema (CUS) muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

El diagrama de casos de uso del sistema se muestra en la siguiente figura:

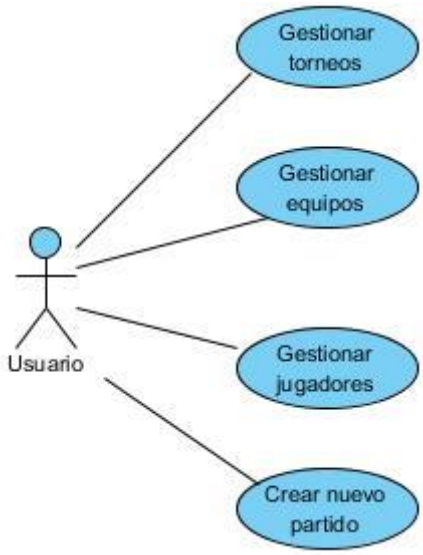


Figura 4: Diagrama de casos de uso del sistema de AndroHB

2.5.3 Descripción de los CUS

A continuación se muestra la descripción del caso de uso Gestionar torneo, la descripción del resto de los casos de uso se pueden consultar en el Anexo 1.

Tabla 2: CUS Gestionar torneos

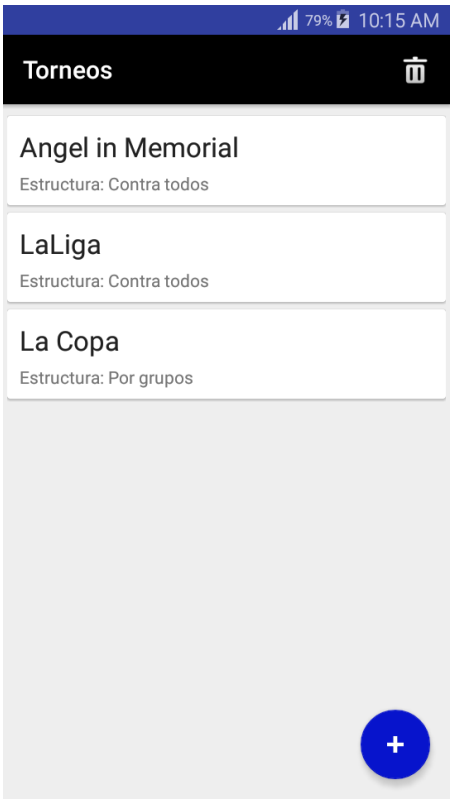
Objetivo	Gestionar los torneos
Actores	Usuario
Resumen	El caso de uso inicia cuando el usuario desea insertar, listar, editar, ver detalles o eliminar torneos
Complejidad	Media
Prioridad	Media
Precondiciones	El usuario selecciona la opción Torneos
Postcondiciones	Se inserta, lista, edita o elimina los torneos

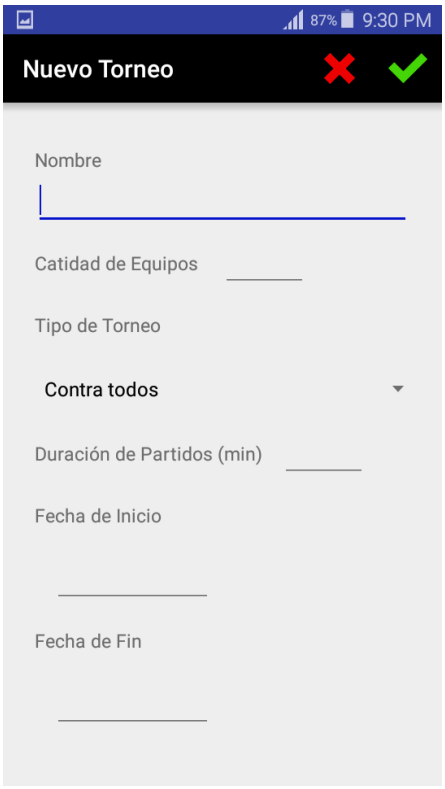
Flujo de eventos		
Flujo básico Gestionar Torneos		
Actor		**Sistema**
Indica que desea insertar, listar, editar, ver detalles o eliminar mínimo un torneo		Si el usuario decide listar los torneos Ver Sección 1: Listar torneos Si el usuario decide insertar un nuevo torneo Ver Sección 2: Insertar torneos Si el usuario decide editar algún torneo Ver Sección 3: Editar torneo Si el usuario decide eliminar torneos Ver Sección 4: Eliminar torneos Si el usuario decide ver los detalles de algún torneo Ver Sección 5: Detalles torneo
Sección 1: Listar torneos		
Flujo básico Listar torneos		
Actor		**Sistema**
1	Selecciona la opción Torneos del menú principal	
2		Muestra una interfaz con una lista de todos los torneos existentes en la base de datos. Las opciones que se brindan son: Insertar un torneo (ver la sección 2 Insertar Torneo) Editar cada torneo (ver la sección 3 Editar torneo) Eliminar torneo (ver la sección 4 Eliminar torneos) Detalles del torneo (ver la sección 5 Detalles

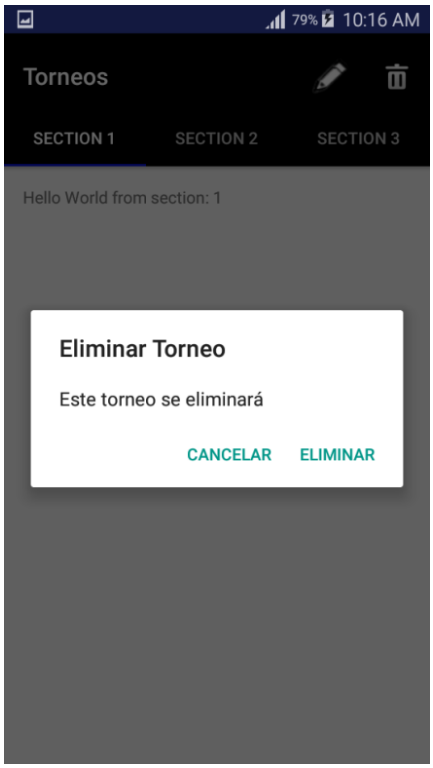
		torneo)
Sección 2 Insertar torneo		
Flujo básico Insertar torneo		
Actor		**Sistema**
1	Selecciona la opción que permite insertar un nuevo torneo	
2		Muestra la interfaz Insertar torneo que permite introducir los valores referentes al torneo a insertar
3	Selecciona el botón Aceptar	
4		Valida los valores introducidos del torneo y muestra un mensaje de información “Torneo Insertado”
Flujo alterno Insertar torneo		
Actor		**Sistema**
1	En caso de hacer clic en el botón Aceptar y haber dejado algún campo en blanco	
2		El sistema muestra el mensaje de error “Todos los campos son obligatorios”
Sección 3 Editar torneo		
Flujo básico Editar torneo		
Actor		**Sistema**
1	Selecciona la opción que permite editar el torneo	

2		Muestra la interfaz Editar torneo que permite editar los valores referentes al torneo
3	Selecciona el botón Aceptar	
4		Valida los nuevos valores introducidos del torneo y muestra un mensaje de información "El Torneo ha sido editado"
Flujo alterno Editar torneo		
Actor		**Sistema**
1	En caso de dar clic en el botón Aceptar y haber dejado algún campo en blanco	
2		El sistema muestra el mensaje de error “Todos los campos son obligatorios”
Sección 4 Eliminar torneo		
Flujo básico Eliminar torneo		
Actor		**Sistema**
1	Selecciona la opción que permite eliminar torneo	
2		Muestra un mensaje de información: “Este torneo se eliminará”
3	Selecciona el botón Confirmar	
4		Elimina el torneo seleccionado y actualiza la lista
Flujo alterno Eliminar torneo		
Actor		

1	Una vez seleccionado el torneo presiona el botón cancelar	
2		Cancela la petición y muestra nuevamente los detalles del torneo.
Sección 5 Detalles torneo		
Flujo básico Detalles torneo		
Actor		**Sistema**
1	Hace clic sobre uno de los torneos listados	
2		Se muestra una interfaz con el listado de los partidos y las estadísticas del torneo.
Relaciones	**CU incluidos**	**-**
Prototipo elemental de interfaz gráfica de usuario		




87% 9:30 PM
Nuevo Torneo
Nombre
Catidad de Equipos
Tipo de Torneo
Contra todos
Duración de Partidos (min)
Fecha de Inicio
Fecha de Fin


79% 10:16 AM
Torneos
SECTION 1
SECTION 2
SECTION 3
Hello World from section: 1
Eliminar Torneo
Este torneo se eliminará
CANCELAR
ELIMINAR

2.6 Arquitectura de software

Según Pressman la arquitectura del software de un programa o sistema de cómputo es la estructura o estructuras del sistema, lo que comprende a los componentes del software, sus propiedades externas visibles y las relaciones entre ellos. Las aplicaciones deben construirse con el empleo de capas en las que se tomen en cuenta distintas preocupaciones; en particular, deben separarse los datos de la aplicación de los contenidos de esta, y estos, a su vez, deben separarse con toda claridad del aspecto y la sensación de la interfaz. (22)

Para el desarrollo del sistema se empleará la arquitectura en tres capas, cuyo principal objetivo es la separación de la lógica de negocios de la lógica de diseño. Esta arquitectura permite llevar a cabo el desarrollo de AndroHB en varios niveles, logrando la reducción del grado de complejidad y facilitando la adaptación a los cambios permitiendo modificar solo el componente necesario sin tener que revisar el código entero. El sistema se divide en las siguientes capas:

- Interfaz: comunica y captura la información proporcionada por el usuario, permite la interacción entre el usuario y el sistema, se evidencia en las interfaces de usuario.
- Lógica de negocio: procesa las peticiones del usuario y se envían las respuestas tras el proceso. Esta capa sirve de intermediaria entre la Interfaz y el Acceso a datos.
- Acceso a datos: es donde se encuentran los datos y es la encargada de acceder a los mismos. Está formada por un gestor de bases de datos que realiza todo el almacenamiento o recuperación de información desde la capa de negocio para proveérsela al usuario.

2.7 Patrón arquitectónico

Los patrones arquitectónicos expresan el esquema de organización estructural fundamental para sistemas de software. Provee un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluye reglas y pautas para la organización de las relaciones entre ellos. (23)

El patrón arquitectónico utilizado para el desarrollo de AndroHB es Modelo-Vista-Controlador (MVC), la cual separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

En la aplicación estos componentes están conformados de la siguiente forma:

- Modelo: recoge y gestiona todos los datos que se manejan en la aplicación, incluye la base de datos de los torneos, los equipos y los jugadores, así como las estadísticas generadas en los partidos.
- Vista: la vista se refiere a los layouts, a lo que el usuario ve en pantalla cuando se ejecuta la aplicación. Estas son el Lenguaje de Marco Extensible (XML por sus siglas del inglés eXtensible Markup Language) en Android.
- Controlador: es la encargada de recibir las órdenes del usuario y se encarga de solicitar los datos a la modelo y de comunicárselo a la vista. Cada vista de la aplicación está asociada a una clase controladora.

2.8 Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicados entre sí, adaptada para resolver un problema de diseño general en un contexto particular. Es una solución para problemas típicos y recurrentes que nos podemos encontrar a la hora de desarrollar una aplicación. Todas las aplicaciones tienen elementos en común, tales como el acceso a datos, la creación de objetos y las operaciones entre sistemas, por esto cuando se desarrolla una nueva, podemos solucionar problemas utilizando algún patrón, ya que son soluciones probadas y documentadas, en lugar de tener que reinventarlo todo desde un principio. (24)

2.8.1 Patrones GRASP

Los patrones GRASP (General Responsibility Assignment Software Patterns o Patrones Generales de Software para Asignación de Responsabilidades) describen los principios fundamentales para asignar responsabilidades a los objetos.

Experto: este patrón responde a la pregunta: ¿cuál es el principio más básico para añadir responsabilidades en una clase? Asigna responsabilidades al experto de la información, es decir, a la clase que tiene la información necesaria para llevar la tarea a cabo. (25)

Se evidencia el patrón en la solución propuesta en las clases Torneo.java, Jugador.java y Equipo.java, donde cada una posee los métodos referentes al tratamiento de la información de los torneos, jugadores y equipos respectivamente (*ver figura 5*).

```java
public class Equipo {
    public final static String ID = "_id";
    public final static String NOMBRE = "nombre";

    @DatabaseField(generatedId = true, columnName = ID)
    private int id;
    @DatabaseField(columnName = NOMBRE)
    private String nombre;

    public int getId() { return id; }
    public void setId(int id) { this.id = id; }

    public String getNombre() {
        return nombre;
    }
    public void setNombre(String nombre) { this.nombre = nombre; }

}
```

Figura 5: Utilización del patrón Experto en la aplicación

Creador: este patrón responde a la pregunta: ¿quién debe ser responsable en la creación de una nueva instancia de una clase? Plantea que debe recaer en la clase que agrega, contiene, registra, utiliza o tiene los datos de inicialización de la nueva instancia. (25)

Ejemplo de este patrón en la aplicación es la clase MainActivity.java (*ver figura 6*).

```java
Button btnInsertarJugador, btnInsertarEquipo, btnListarJugador, btnListarEquipo;
btnInsertarJugador = (Button) findViewById(R.id.btnInsertarJugador);
btnInsertarEquipo = (Button) findViewById(R.id.btnInsertarEquipo);
btnListarJugador = (Button) findViewById(R.id.btnListarJugador);
btnListarEquipo = (Button) findViewById(R.id.btnListarEquipo);
```

Figura 6: Utilización del patrón Creador en la aplicación

Bajo Acoplamiento: este patrón responde a la pregunta: ¿cómo soportar baja dependencia e incrementar la reutilización? Propone asignar responsabilidades de tal manera que el acoplamiento sea el menor posible. (25)

La (*Figura 7*) muestra cómo fue empleado el patrón bajo acoplamiento en la implementación de la aplicación.

```
public class MainJugadores extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_jugadores);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener((view) → {
                Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
        });
    }

}
```

Figura 7: Utilización del patrón Bajo Acoplamiento en la aplicación

Alta Cohesión: este patrón responde a la pregunta: ¿cómo lograr que la complejidad sea lo más manejable posible? Propone asignar responsabilidades procurando que la cohesión sea lo más alta posible. (25)

La (*Figura 8*) muestra cómo fue empleado el patrón alta cohesión en la implementación de la aplicación.

```java
public class Jugador {

    public final static String ID = "_id";
    public final static String NOMBRE = "nombre";
    public final static String APELLIDO = "apellido";
    public final static String FECHA_NACIMIENTO = "fecha_nacimiento";
    public final static String EQUIPO = "equipo";

    @DatabaseField(generatedId = true, columnName = ID)
    private int id;
    @DatabaseField(columnName = NOMBRE)
    private String nombre;
    @DatabaseField(columnName = APELLIDO)
    private String apellido;
    @DatabaseField(columnName = FECHA_NACIMIENTO)
    private String fechaNacimiento;
    @DatabaseField(foreign = true, columnName = EQUIPO)
    private Equipo equipo;

    public int getId() { return id; }
    public void setId(int id) { this.id = id; }

    public String getNombre() { return nombre; }
    public void setNombre(String nombre) { this.nombre = nombre; }

    public String getApellido() { return apellido; }
    public void setApellido(String apellido) { this.apellido = apellido; }

    public String getFechaNacimiento() { return fechaNacimiento; }
    public void setFechaNacimiento(String fechaNacimiento) {
        this.fechaNacimiento = fechaNacimiento;
    }
```

Figura 8: Utilización del patrón Alta Cohesión en la aplicación

Controlador: este patrón responde a la pregunta: ¿quién debe manejar eventos del sistema? Propone asignar responsabilidades para el manejo de mensajes de eventos del sistema a una clase que represente al conjunto del sistema o negocio, represente algo del mundo real que está activo, o representa un administrador artificioso para todos los eventos del sistema. (25)

Un ejemplo de este patrón en la aplicación se puede ver en la (*figura 9*).

```
public void onClick(View v) {

    Log.v("abc", "" + v);
    if(v == btnInsertarJugador) {
        startActivity(new Intent(this, InsertarJugador.class));
    }
    else if(v == btnInsertarEquipo) {
        startActivity(new Intent(this, InsertarEquipo.class));
    }
    else if(v == btnListarJugador) {
        startActivity(new Intent(this, ViewStudentRecordActivity.class));
    }
    else if(v == btnListarEquipo) {
        startActivity(new Intent(this, ViewTeacherRecordActivity.class));
    }
}
```

Figura 9: Utilización del patrón Controlador en la aplicación

2.8.2 Patrones GoF

Los patrones GoF (Gang of Four, en español: Pandilla de Cuatro) se dividen en tres tipos: patrones de creación, de estructura y de comportamiento. A continuación se muestran los empleados en AndroHB.

Singleton: el objetivo de este patrón es asegurarse de que de una clase solo existe una instancia y que esta es accesible. (26)

En el siguiente fragmento *(figura 10)* de código se evidencia este patrón dentro de AndroHB.

```
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK && event.getRepeatCount() == 0) {
        Intent intent = new Intent(insertar_equipo.this, MainEquipos.class);
        startActivity(intent);
        finish();
        return true;
    }
    return super.onKeyDown(keyCode, event);
}
```

Figura 10: Utilización del patrón Singleton en la aplicación

Decorador: permite añadir funcionalidades o responsabilidades a un objeto de forma dinámica, propone que cuando cambie el estado de un objeto sean actualizados de forma automática todos los otros objetos que dependen de este. (26) En la aplicación se utiliza este patrón cuando se introducen las acciones de un partido se van actualizando automáticamente las estadísticas referentes al torneo que se esté jugando así como las de los equipos y jugadores.

Facade: propone crear una interfaz unificada para manejar un conjunto de objetos en un subsistema. De esta forma permite que los usuarios que utilicen el sistema no necesiten conocer los diferentes subsistemas y su funcionamiento, sino que solo necesitan conocer la interfaz para manejar el sistema completo. (26)

En la (*figura 11*) se evidencia este patrón en la aplicación.

```
public class insertar_jugador extends AppCompatActivity implements AdapterView.OnItemSelectedListener {
    Spinner posicion;
    String [] cadena = {"Jugador de Campo", "Portero"};
    ImageView foto;
    private EditText nombre_jugador;
    private EditText apellidos_jugador;

    private static final int REQUEST_PHOTO = 1;
    private static final int RESULT_LOAD_IMAGE = 0;
    private String picturePath;
    private Boolean camera;
    private Jugador jugador = new Jugador();
    private DBHelper dbhelper = null;
    private String name = "";
    private Bitmap esta;
```

Figura 11: Utilización del patrón Facade en la aplicación

2.9 Modelo de datos

El modelado de datos permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades. El modelo de AndroHB (ver *figura 12*) ofrece una descripción abstracta sobre la representación de los datos utilizados en la aplicación. Este modelo contiene características propias de estos objetos y la forma en que están relacionados.

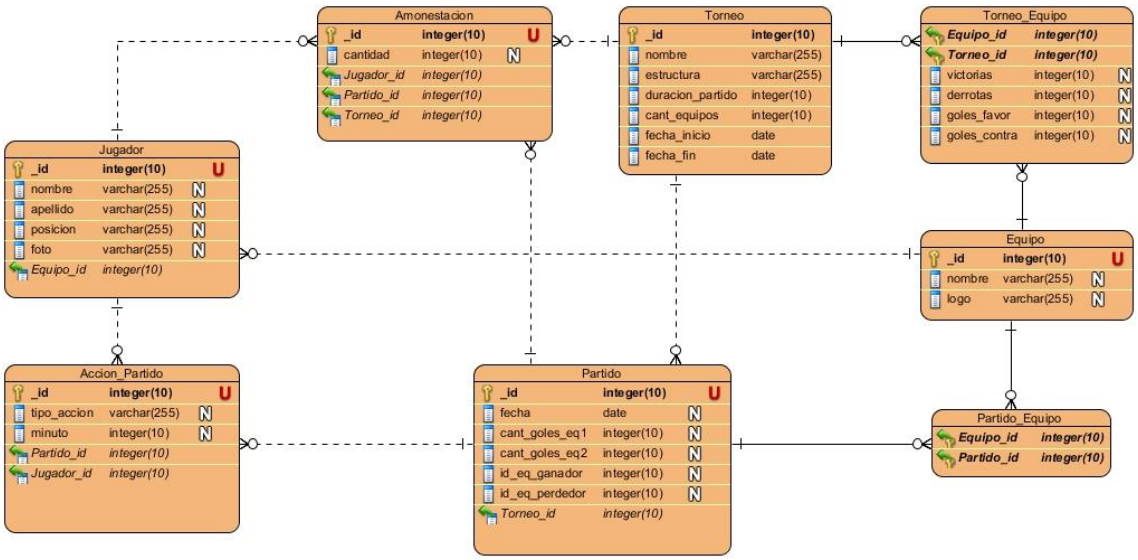


Figura 12: Modelo físico de la base de datos.

Conclusiones del capítulo

En este capítulo fue descrita la propuesta de solución. Se definieron los artefactos correspondientes a las etapas de análisis y diseño de la aplicación AndroHB, según la metodología AUP-UCI. Se identificaron y describieron los requisitos funcionales y no funcionales del sistema. Se definió como arquitectura de software a emplear la Arquitectura en tres Capas y como patrón arquitectónico el MVC, logrando así una aplicación mucho más organizada y menos compleja a la hora de realizar alguna modificación. Se identificaron además los patrones de diseño GRASP y GoF a utilizar en el desarrollo de la solución y se diseñó la base de datos.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DE LA SOLUCIÓN PROPUESTA

En el presente capítulo se describen los elementos requeridos para desarrollar AndroHB a partir de los resultados obtenidos de las etapas de análisis y diseño. Se muestran el diagrama de componentes, diagrama de despliegue, diagrama de paquetes y se definen los estándares de codificación utilizados para el desarrollo de la aplicación. Se precisan las pruebas de software a aplicar y se muestran los resultados obtenidos.

3.1 Modelo de implementación

El Modelo de Implementación comprende un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes podemos encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos. (27)

3.1.1 Diagrama de componentes

Un diagrama de componentes permite visualizar la estructura de alto nivel del sistema y el comportamiento del servicio que estos componentes proporcionan y usan a través de interfaces. Además, se utilizan para describir un diseño que está implementado en cualquier idioma o estilo. Solo es necesario identificar los elementos del diseño que interactúan con los otros elementos del mismo a través de un conjunto restringido de entradas y salidas. (28)

A continuación se muestra el diagrama de componentes de AndroHB (*figura 13*):

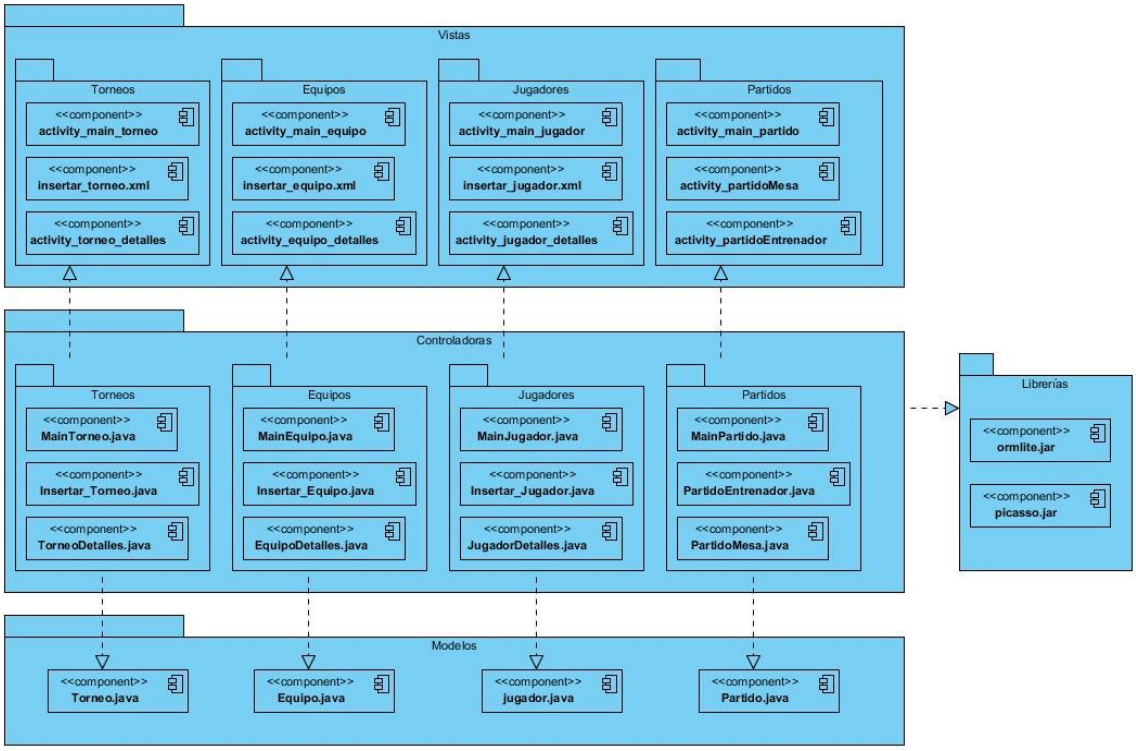


Figura 13: Diagrama de Componentes

3.2 Estándares de codificación

Los estándares de codificación son un conjunto de directrices, normas y reglamentos que se establecen especificando como debe escribirse el código fuente de una aplicación para lograr una organización en el mismo. A continuación se listan los elementos del estándar de codificación definido para el desarrollo de AndroHB:

- Dejar espacio de separación dentro del código, para que sea entendible.
- El nombre de todas las variables y métodos comenzarán con letra minúscula y si este está compuesto por varias palabras se utilizará el estilo de escritura lowerCamelCase.
- Los nombres de las variables deben ser cortos y significativos.
- En el contenido siempre se identará con tabulaciones, nunca utilizando espacios en blanco.
- El nombre de las clases comenzará con mayúsculas y cada salto de palabras debe iniciar con mayúsculas.
- Cuando una expresión exceda la línea, se romperá después de una coma o antes de un operador.

3.3 Diagrama de despliegue

Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos. (29)

A continuación se muestra el diagrama de despliegue de la aplicación, ver (figura 14):

Figura 14: Diagrama de despliegue

Dispositivo Móvil: representa el hardware donde se ejecutará el sistema. La base de datos que utiliza el mismo es interna y estará representada por el fichero androhb.db, el cual almacenará toda la información con que trabaja la aplicación. El sistema operativo del dispositivo debe ser Android. Este debe contar con los requisitos no funcionales especificados en el acápite 2.4.

3.4 Diagrama de paquetes

El diagrama de paquetes se utiliza para modelar los paquetes que componen una aplicación. Un paquete es una parte de un modelo que contiene elementos al más alto nivel, tales como las clases y sus relaciones. Cada uno puede contener otros paquetes y las dependencias entre ellos resumen dependencias entre los elementos internos a ellos. (30)

En la siguiente (*figura 15*) se puede apreciar el diagrama de paquetes de AndroHB:

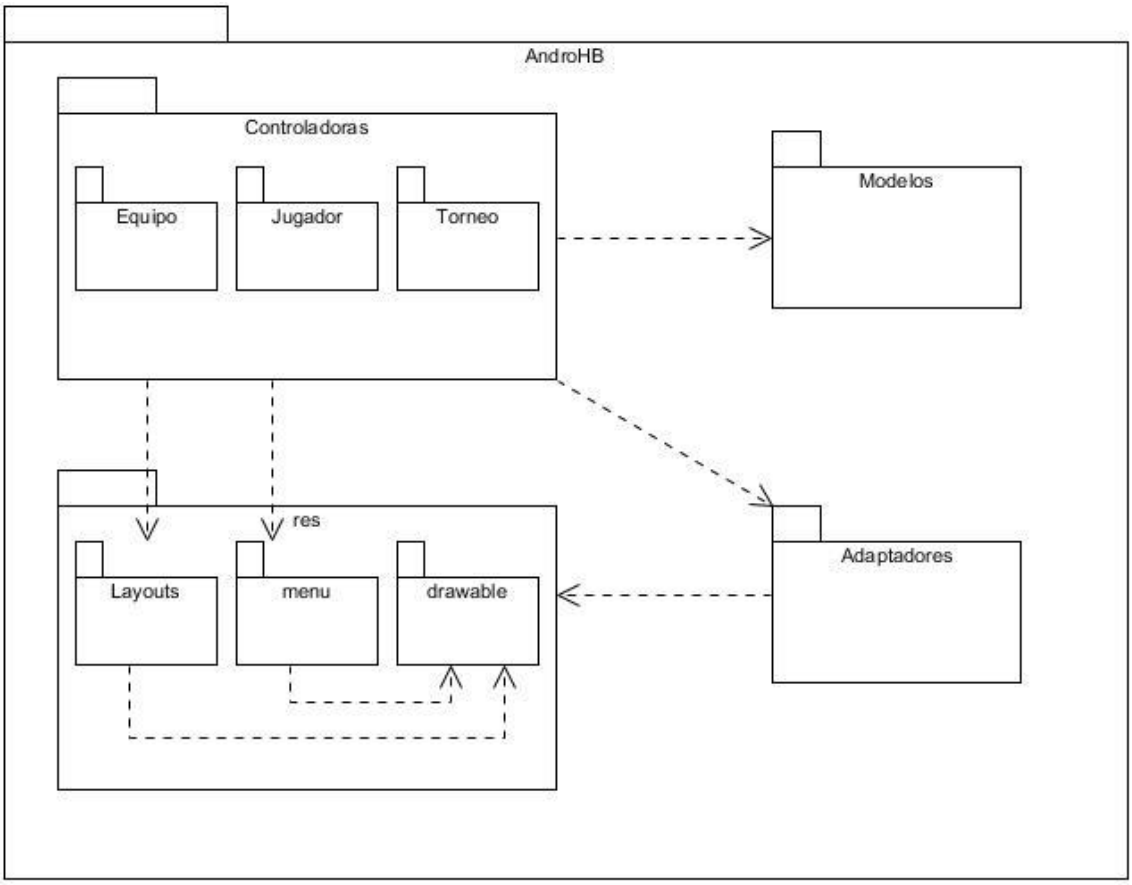


Figura 15: Diagrama de paquetes

A continuación se muestra la descripción de los paquetes:

Controladoras: contiene todas las clases controladoras de la aplicación, contiene dentro tres paquetes donde cada uno almacena distintas clases controladoras.

Equipo: almacena las clases controladoras referentes a la gestión de los equipos.

Jugador: almacena las clases controladoras referentes a la gestión de los jugadores.

Torneo: almacena las clases controladoras referentes a la gestión de los torneos.

Modelos: contiene las clases entidades que se utilizan como clases de acceso a datos.

Adaptadores: contiene las clases que sirven como adaptadores entre las clases controladoras y los layouts para poder realizar las acciones de listar equipos, jugadores, torneos y partidos en la aplicación.

Res: almacena otros paquetes en los cuales se almacenan los elementos que componen la interfaz de la aplicación.

Layouts: contiene todos los archivos XML que conforman las vistas de la aplicación.

Menú: contiene los archivos XML que componen todos los menús con que cuenta la aplicación.

Drawable: contiene todas las imágenes que son cargadas desde los archivos XML de la aplicación.

3.5 Pruebas de software

Actualmente, desarrollar un software sin fallos y que satisfaga totalmente las necesidades de los clientes es la principal preocupación de los equipos de desarrollo. Aquí surge el término de calidad de software, el cual está definido por "*la totalidad de características de un producto, proceso o servicio que cuenta con la habilidad de satisfacer necesidades explícitas o implícitas"* (31). Para garantizar la misma se realizan las distintas pruebas de software. A continuación se exponen las pruebas realizadas a AndroHB.

3.5.1 Estrategia de prueba

Una estrategia de prueba de software proporciona una guía que describe los pasos que deben realizarse como parte de la prueba, cuándo se planean y se llevan a cabo dichos pasos, y cuánto esfuerzo, tiempo y recursos se requerirán. Por tanto, cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de casos de prueba, la ejecución de la prueba y la recolección y evaluación de los resultados. (22)

Según Pressman existen cuatro niveles de prueba:

- Pruebas unitarias
- Pruebas de integración
- Pruebas de validación
- Pruebas del sistema

Las pruebas a AndroHB serán realizadas en dos de los cuatro niveles anteriormente definidos, en el de pruebas unitarias y en el de pruebas del sistema.

Para el nivel de pruebas unitarias se realizarán pruebas de caja blanca, las cuales se basan en el examen cercano de los detalles de procedimiento. Las rutas lógicas a través del software y las colaboraciones entre componentes se ponen a prueba al revisar conjuntos específicos de condiciones y/o bucles. (22)

Por otro lado en el nivel de pruebas del sistema serán empleadas las pruebas de caja negra, que son las que se llevan a cabo en la interfaz del software. Una prueba de caja negra examina algunos aspectos fundamentales de un sistema con poca preocupación por la estructura lógica interna del software. (22)

3.5.2 Pruebas unitarias

Las pruebas unitarias se focalizan en ejecutar cada módulo, lo que provee un mejor modo de manejar la integración de las unidades en componentes mayores. Busca asegurar que el código funciona de acuerdo con las especificaciones y que el módulo lógico es válido. (32)

La aplicación se dividió en los siguientes cuatro módulos para realizar las pruebas unitarias:

1. Partido: agrupa todas las clases referentes a los partidos.
2. Torneo: agrupa todas las clases referentes a los torneos.
3. Equipo: agrupa todas las clases referentes a los equipos.
4. Jugador: agrupa todas las clases referentes a los jugadores.

A estos módulos se le aplicaron pruebas de caja blanca directas al código utilizando el método del camino básico, el cual “permite obtener una medida de la complejidad de un diseño procedimental, y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez” (33).

A continuación se muestra una prueba de caja blanca realizada al método onActivityResult de la clase Insertar_Jugador.java del módulo Jugador:

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (resultCode == Activity.RESULT_OK) {
        Bundle extras;
        switch (requestCode) {
            case REQUEST_PHOTO:
                if((data != null) && (data.hasExtra("data"))){
                        foto.setImageBitmap((Bitmap) data.getParcelableExtra("data"));
                        extras = data.getExtras();
                        esta = (Bitmap) extras.get("data");
                }else{
                    foto.setImageBitmap(BitmapFactory.decodeFile(name));}
                camera=true;
                flag=true;
                break;
            case RESULT_LOAD_IMAGE:
                if (data != null) {
                    Uri selectedImage = data.getData();
                    String[] filePathColumn2 = {MediaStore.Images.Media.DATA};
                    Cursor cursor2 = getContentResolver().query(selectedImage, filePathColumn2, null, null, null);
                    cursor2.moveToFirst();
                    int columnIndex2 = cursor2.getColumnIndex(filePathColumn2[0]);
                    picturePath = cursor2.getString(columnIndex2);
                    cursor2.close();
                    foto.setImageURI(selectedImage);
                    camera = false;}
                flag=true;
                break;}
        super.onActivityResult(requestCode, resultCode, data);}
}
```

Figura 16: Método onActivityResult

Así queda el grafo de flujo de este método:

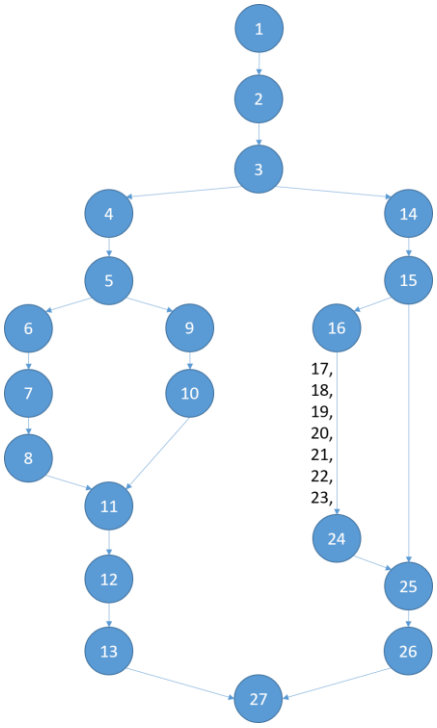


Figura 17: Grafo de flujo del método onActivityResult

La complejidad ciclomática, V(G), del grafo se calcula con la fórmula:

V(G) = Aristas - Nodos + 2

V(G) = 29 – 27 + 2

V(G) = 4

A partir de la complejidad ciclomática se obtiene la cantidad de caminos independientes con que cuenta el grafo, en este caso son los siguientes cuatro:

- 1,2,3,4,5,6,7,8,11,12,13,27
- 1,2,3,4,5,9,10,11,12,13,27
- 1,2,3,14,15,16,17,18,19,20,21,22,23,24,25,26,27
- 1,2,3,14,15,25,26,27

Partiendo de esto se diseñaron cuatro casos de pruebas, uno para cada camino y se probaron con el debugueo de la aplicación.

Los casos de prueba son los siguientes:

1. Variable *requestCode* = “REQUEST_PHOTO” y variable *data* = null.
2. Variable *requestCode* = “REQUEST_PHOTO”, variable *data* != null, con extras pasados a través del Bundle *extras*.
3. Variable *requestCode* = “RESULT_LOAD_IMAGE” y variable *data* = null.
4. Variable *requestCode* = “RESULT_LOAD_IMAGE” y variable *data* != null.

Después de realizada la prueba se obtuvo un resultado satisfactorio en el que no se detectaron No Conformidades (NC). En pruebas de este tipo realizadas al resto de los módulos de la aplicación se detectaron dos NC. La primera NC (NC1) se detectó en el método setOnChronometerTickListener de la clase partido_entrenador.java del módulo Partido, la NC detectada consistía en un error de objeto con referencia nula de las instancias de la clase Cronometro.java. Se le dio solución creando condicionales para cada uno de los objetos de la clase Cronometro.java (*ver figura 18*).

La segunda NC (NC2) se detectó en el método onKeyDown de la clase insertar_jugador.java del módulo Jugador, la NC consistía en que si se accedía a editar el jugador desde el listado de jugadores de un equipo, al presionar la tecla de ir hacia atrás, entraba en la condicional de ir hacia el listado general de jugadores, por lo que direccionaba hacia la actividad incorrecta. El problema fue solucionado enviando un Bundle desde la clase detalles_equipo.java que incluye una variable booleana que indica desde donde fue llamada la actividad (ver *figura 19*).

```
crono.setOnChronometerTickListener((chronometer) → {
        if (!prorroga) {
            if (crono.getText().toString().equalsIgnoreCase(tiempo / 2 + ":00")
                    || crono.getText().toString().equalsIgnoreCase("0" + tiempo / 2 + ":00")) {
                timeWhenStopped = crono.getBase() - SystemClock.elapsedRealtime();
                crono.stop();
                if(cronometro1 != null)
                    if (cronometro1.getIniciado()){
                        cronometro1.pause();
                    }
                if(cronometro2 != null)
                    if(cronometro2.getIniciado()) {
                        cronometro2.pause();
                    }
                if(cronometro3 != null)
                    if (cronometro3.getIniciado()) {
                        cronometro3.pause();
                    }
                MediaPlayer mp = MediaPlayer.create(partido_entrenador.this, R.raw.sirena_short);
                mp.start();
                if (!bandera) {
                    menu.findItem(R.id.action_play).setIcon(R.drawable.play);
                    Toast.makeText(partido_entrenador.this,
                            "Fin del primer tiempo", Toast.LENGTH_SHORT).show();
                    enable = false;
                } else {
                    enable = true;
                }
                iniciado = false;
                bandera = true;
            }
```

Figura 18: Método setOnChronometerTickListener

```
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK && event.getRepeatCount() == 0) {
        Bundle bl = this.getIntent().getExtras();
        if (bl != null){
            if(nameTeam != null) {
                try {
                    List<Equipo> nE = equipoDao.queryForEq(Equipo.NOMBRE, nameTeam);
                    Intent intent = new Intent(insertar_jugador.this, EquipoDetalles.class);
                    Bundle b = new Bundle();
                    b.putInt("id", nE.get(0).getId());
                    intent.putExtras(b);
                    startActivity(intent);
                    finish();
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }else if (bl.getBoolean("eq") ){
                //Si se entro a editar desde el listado de jugadores de un equipo
                Intent intent = new Intent(insertar_jugador.this, JugadorDetalles.class);
                Bundle b = new Bundle();
                b.putBoolean("eq", true);
                b.putInt("id", jugador_a_editar.get(0).getId());
                intent.putExtras(b);
                startActivity(intent);
                finish();
```

Figura 19: Método onKeyDown

3.5.3 Pruebas al sistema

Las pruebas al sistema se realizan para detectar fallas en el cubrimiento de los requerimientos. Estas pruebas serán realizadas a AndroHB desde un dispositivo móvil. Para llevar a cabo las mismas se hace necesario el diseño de casos de prueba que indiquen el comportamiento que debe tener la aplicación ante determinado escenario para compararlo con el comportamiento real obtenido en el mismo.

A continuación se pueden apreciar algunos de los casos de prueba que fueron diseñados para realizar estas pruebas de caja negra al sistema, el resto puede ser consultado en el Anexo 2.

Tabla 3: Caso de prueba Listar jugadores

Escenario	**Descripción**	**Respuesta del sistema**	**Flujo central**
EC 1.1 Listar Jugadores	Permite al usuario listar los jugadores creados.	Muestra una lista de todos los jugadores creados por el usuario.	1- Seleccionar del menú principal la opción "Jugadores"

Tabla 4: Caso de prueba Eliminar jugador

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.2 Eliminar Jugador	Permite al usuario eliminar uno de los jugadores creados.	Elimina el jugador seleccionado por el usuario.	1- Seleccionar del menú principal la opción "Jugadores" 2- Seleccionar el jugador que desea eliminar 3- Seleccionar la opción eliminar de la barra superior 4- Confirmar la eliminación

Tabla 5: Caso de prueba Listar equipos

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 2.1 Listar Equipos	Permite al usuario listar los equipos creados.	Muestra una lista de todos los equipos creados por el usuario.	1- Seleccionar del menú principal la opción "Equipos"

Tabla 6: Caso de prueba Eliminar equipo

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 2.2 Eliminar Equipo	Permite al usuario eliminar uno de los equipos creados.	Elimina el equipo seleccionado por el usuario.	1- Seleccionar del menú principal la opción "Equipos" 2- Seleccionar el equipo que desea eliminar 3- Seleccionar la opción eliminar de la barra superior 4- Confirmar la eliminación

Después de realizadas las pruebas al sistema fueron detectadas seis NC:

- NC3: al listar, ya fueran equipos o jugadores, a partir de cierto número de elementos en la lista comenzaba a repetir las imágenes de los que primero se guardaron en la base de datos.
- NC4: no se mostraban los resultados de los partidos en los detalles de los torneos.
- NC5: el sistema no actualizaba las tablas de las estadísticas durante el partido.
- NC6: el sistema permitía seguir introduciendo acciones aunque el tiempo del partido se encontrará detenido.
- NC7: los cronómetros de las amonestaciones no se detenían al terminar un período del juego.
- NC8: presentaba problemas en la navegación.

Las NC detectadas fueron solucionadas en las iteraciones de las pruebas (ver *figura 20*). Además, se probó la solución propuesta en varias ocasiones, formando el cliente una parte significativa de las pruebas y obteniéndose así un resultado satisfactorio.

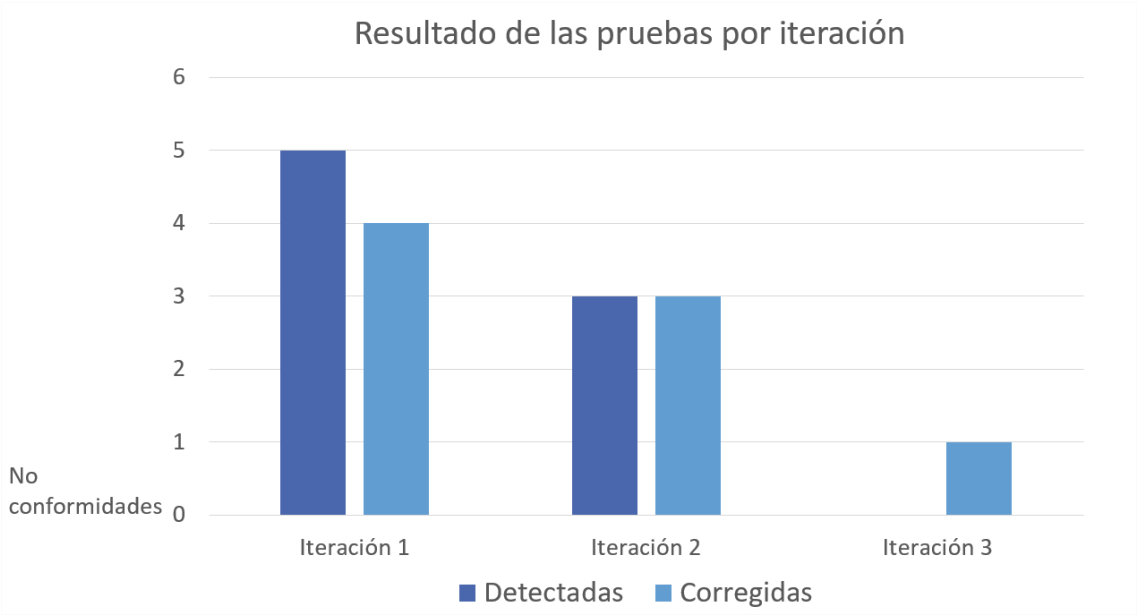


Figura 20: Resultado de las pruebas por iteración

A continuación se muestran en la (*tabla 7*) las no conformidades detectadas en cada una de las iteraciones de las pruebas realizadas.

Tabla 7: Resultado de las pruebas

Número	Iteración	Observación
NC1	1	Detectada y corregida en la iteración 1

NC2	1	Detectada y corregida en la iteración 1
NC3	1	Detectada y corregida en la iteración 1
NC4	1, 2, 3	Detectada en la iteración 1 y corregida en la iteración 3
NC5	1	Detectada y corregida en la iteración 1
NC6	2	Detectada y corregida en la iteración 2
NC7	2	Detectada y corregida en la iteración 2
NC8	2	Detectada y corregida en la iteración 2

Además de las pruebas anteriormente descritas, la aplicación fue probada en dispositivos móviles con diferentes dimensiones (*ver tabla 8*) para asegurar que fuese adaptable a cualquier pantalla, garantizando así un diseño adaptativo.

Tabla 8: Visualización de la aplicación en distintos dispositivos

Dispositivo Móvil	Tamaño de la pantalla	Resolución de la pantalla	
Alcatel One Touch 5036A	4.5" pulgadas	480 x 854 píxeles	
BLU Dash X2	5" pulgadas	720 x 1280 píxeles	
Tablet ZTE E10T	10.1" pulgadas	720 x 1280 píxeles	

Conclusiones del Capítulo

En este capítulo fue descrito el proceso de implementación de AndroHB a través los artefactos utilizados para llevar a cabo la misma; los diagramas de componentes, despliegue y paquetes. Además, se expuso el estándar de codificación empleado en la escritura del código fuente del sistema. Se presentó la estrategia de prueba tomada por el equipo de desarrollo, que comprendió la realización de las pruebas unitarias y las pruebas al sistema como métrica para erradicar los fallos y medir el correcto funcionamiento y eficiencia del producto.

CONCLUSIONES GENERALES

Al finalizar el presente trabajo de diploma, se cumple con el objetivo general planteado al inicio de la investigación, por lo que se puede arribar a las siguientes conclusiones:

1. Con el estudio de las aplicaciones de apoyo a entrenadores de balonmano existentes a nivel nacional e internacional, se comprobó que ninguna cumple con las necesidades del cliente en su totalidad, por lo que se hizo necesario el desarrollo de AndroHB.
2. La selección de las herramientas, tecnologías y la metodología de desarrollo de software AUP-UCI, permitió el desarrollo de un producto acorde con las necesidades establecidas por el cliente.
3. Se diseñó una solución muy práctica y sencilla para la recogida de información durante los partidos de balonmano, agilizando así este proceso, permitiendo además el cálculo automático de estadísticas y la gestión de jugadores, equipos y torneos.
4. El software AndroHB fue desarrollado haciendo uso de estándares y buenas prácticas de programación de acuerdo a lo descrito en el acápite 3.2 de la investigación, garantizando limpieza y organización en el código del sistema.
5. Para validar la calidad del software AndroHB fueron realizadas las pruebas de caja blanca y caja negra, obteniéndose resultados satisfactorios tras la corrección de los errores existentes en cuatro iteraciones.

RECOMENDACIONES

Los autores de la presente investigación recomiendan:

- Añadir para próximas iteraciones o versiones del producto, funcionalidades que le permita intercambiar entre varios dispositivos móviles, la información recogida.
- Extender el uso de este tipo de aplicación para otros deportes como voleibol o fútbol.

REFERENCIAS BIBLIOGRÁFICAS

1. *"Android" el sistema operativo de Google para dispositivos móviles.* **Kristel Malave Polanco, Jose Luis Beauperthuy Taibo.** 19, s.l. : Negotium, 2011.

2. Tácticas del balonmano. [En línea] http://www.cheesehosting.com/balonmano/tecnicas-de-balonmano/tacticas-del-balonmano.php.

3. Handball Manager 12. [En línea] https://play.google.com/store/apps/details?id=com.valeriologiudice.assistantcoachhandball.

4. Assistant Coach Handball. [En línea] https://play.google.com/store/apps/details?id=com.valeriologiudice.assistantcoachhandball.

5. Estadísticas Balonmano. [En línea] https://play.google.com/store/apps/details?id=com.whattheappz.handball.

6. Handball Board. [En línea] https://play.google.com/store/apps/details?id=com.mokyn.sportsboard.HB.

7. Estadisticas Handball. [En línea] https://play.google.com/store/apps/details?id=com.aat.estadisticasHandball&hl=es.

8. Scoreboard Handball. [En línea] https://play.google.com/store/apps/details?id=it.alecs.puntihandball&hl=es.

9. **María de los Angeles Obregón Valdés, Heimer Lambert Delgado.** *Sistema informático para el análisis del comportamiento de acciones tácticas significativas de los equipos de balonmano .* 2010.

10. **Jose Luis Cendejas Valdez.** eumed.net - Enciclopedia Virtual. *Modelos y Metodologías.* [En línea] 8 de 05 de 2014. http://www.eumed.net/tesis-doctorales/2014/jlcv/software.htm.

11. **Acuña, Karenny Brito.** *SELECCION DE METODOLOGÍAS DE DESARROLLO PARA APLICACIONES.* 2012.

12. *Aplicando el método de Boehm y Turner.* **Mairelys Boeras Velázquez, Laritza Cabrera Barroso.** La Habana : Ediciones Futuro, 19 de 03 de 2012.

13. **Sintya Milena Meléndez Valladarez, Maria Elizabeth Gaitan, Neldin Noel Pérez Reyes.** *Métodología ágil de desarrollo de software Programación Extrema.* 2016.

14. **Sánchez, Tamara Rodríguez.** *Metodología de desarrollo para la actividad prodcutiva de la UCI.* La Habana, Cuba : s.n., 2015.

15. *CONCEPTOS FUNDAMENTALES DE LA PROGRAMACIÓN ORIENTADA A OBJETOS.* **Mariano, Andrea Melendres.** 2014.

16. Unified Modeling Language - Sitio Oficial. [En línea] http://www.uml.org.

17. Android Studio. *Todo lo que necesitas para realizar compilaciones en Android.* [En línea] https://developer.android.com/studio/features.html?hl=es-419.

18. SQLite. *About SQLite.* [En línea] http://www.sqlite.org/about.html.

19. Academia Android. *Introducción a ORM y GreenDAO.* [En línea] http://academiaandroid.com/introduccion-a-orm-y-greendao/.

20. Visual Paradigm para UML. [En línea] http://www.software.com.ar/p/visual-paradigm-para-uml.

21. *Modelo del Dominio.* **Larman, Craig.** 2003.

22. **Pressman, Roger S.** *Ingeniería de Software - Un Enfoque Práctico - 7ma Edición.*

23. *Arquitecturas de Software - Guía de Estudio.* **Erika Camacho, Fabio Cardeso, Gabriel Núñez.** 2004.

24. Patrones de diseño: qué son y por qué debes usarlos. *Genbeta:dev.* [En línea] 14 de 07 de 2014. https://www.genbetadev.com/metodologias-de-programacion/patrones-de-diseno-que-son-y-por-que-debes-usarlos.

25. *Patrones de diseño.* s.l. : Ingeniería del Software II 2011, 2011. Tema 6. Patrones de diseño.

26. Mundo Informático. *Patrones de Diseño GoF.* [En línea] https://infow.wordpress.com/category/patrones-de-disenogof/.

27. MODELO DE IMPLEMENTACIÓN. [En línea] 01 de 06 de 2013. http://ithleovi.blogspot.com/2013/06/unidad-5-modelo-deimplementacion-el.html.

28. Microsoft - Developer Network. *Diagramas de componentes de UML: Referencia.* [En línea] https://msdn.microsoft.com/es-es/library/dd409390.aspx.

29. Sparx Systems. *Diagrama de Despliegue UML 2.* [En línea] http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.

30. **Armando Rosales Valvidea, Moises Cruz Jose.** SlideShare. *Diagrama de Paquetes.* [En línea] https://es.slideshare.net/moyscruz/diagramas-de-paquetes.

31. eumet.net - Enciclopedia Virtual. *MEJORES PRÁCTICAS PARA EL ESTABLECIMIENTO Y ASEGURAMIENTO DE LA CALIDAD DE SOFTWARE .* [En línea] http://www.eumed.net/libros-gratis/2008a/351/Calidad%20de%20Software.htm.

32. **Londoño, Jorge Hernan Abad.** Ingeniería de Software. *Tipos de Prueba de software.* [En línea] 6 de 04 de 2005. http://ing-sw.blogspot.com/2005/04/tipos-de-pruebas-de-software.html.

33. **Giraldo, Melissa.** Prezi. *Camino Básico.* [En línea] 9 de 04 de 2014. https://prezi.com/jvbhsculhy2k/camino-basico/?webgl=0.

34. **Carolina Martínez, Ivette.** *Modelo Conceptual/Modelo de Dominio.* [Documento] Caracas : Universidad Simón Bolívar, 2010.

35. *Taller sobre artefactos ingenieriles que deben generar de acuerdo a la metodología seleccionada y las necesidades del proyecto.*

36. Android OS. *Características.* [En línea] http://androidos.readthedocs.io/en/latest/data/caracteristicas/.

37. Reutilización del Software - Patrones de Diseño. [En línea] http://siul02.si.ehu.es/~alfredo/iso/06Patrones.pdf.

38. *Patrones GRASP.* **Usaola, Macario Polo.** Macario Polo Usaola -Patrones GRASP. pág. 18.

39. **Erich Gamma, Ralph Johnson, John Vlissides, Richard Helm.** *Design Patterns.* s.l. : Addison-Wesley, 1994.

40. *Metodología de desarrollo para la actividad prodcutiva de la UCI.*

ANEXOS

Anexo 1: Descripción de los CUS

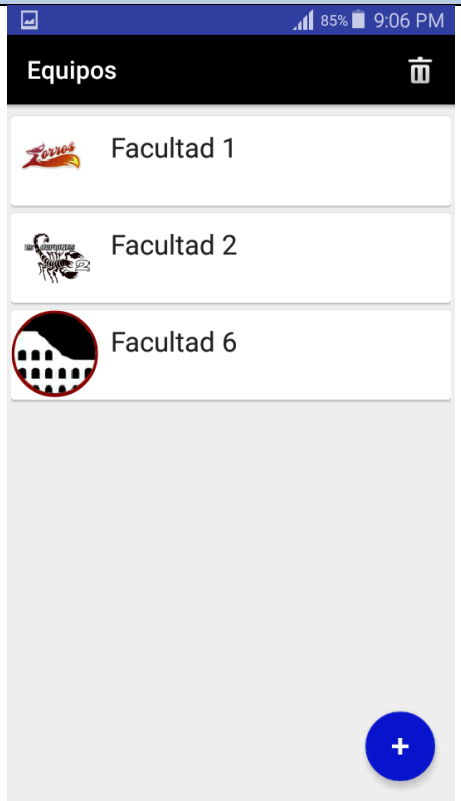
Tabla 9: CUS Gestionar equipos

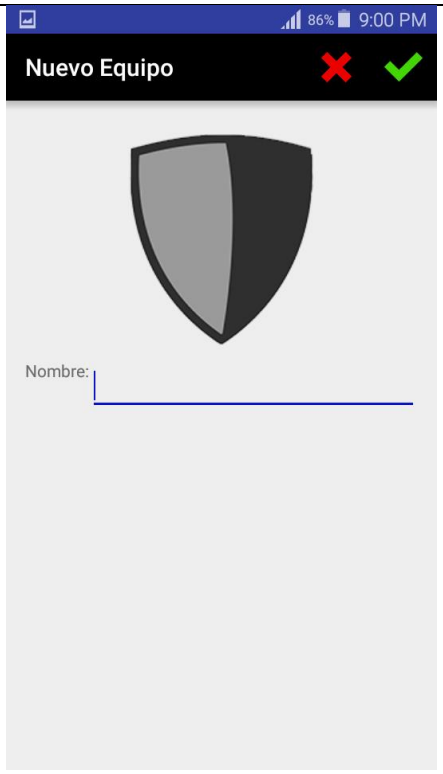
Objetivo	Gestionar los equipos	
Actores	Usuario	
Resumen	El caso de uso inicia cuando el usuario desea insertar, listar, editar, ver detalles o eliminar equipos	
Complejidad	Media	
Prioridad	Media	
Precondiciones	El usuario selecciona la opción equipos	
Postcondiciones	Se inserta, lista, edita o elimina los equipos	
Flujo de eventos		
Flujo básico Gestionar equipo		
Actor		**Sistema**
Indica que desea insertar, listar, editar, ver detalles o eliminar un equipo		Si el usuario decide listar los equipos Ver Sección 1: Listar equipos Si el usuario decide insertar un nuevo equipo Ver Sección 2: Insertar equipo Si el usuario decide editar algún equipo Ver Sección 3: Editar equipo Si el usuario decide eliminar equipo Ver Sección 4: Eliminar equipo Si el usuario decide ver los detalles de algún equipo Ver Sección 5: Detalles equipo Si el usuario decide exportar las estadísticas a PDF. Ver Sección 6: Exportar estadísticas a PDF
Sección 1: Listar equipos		
Flujo básico Listar equipos		
Actor		**Sistema**
1	Selecciona la opción Equipos del menú principal	
2		Muestra una interfaz con una lista de todos los equipos existentes en la base de datos. Las opciones que se brindan son:

		Insertar un equipo (ver la sección 2 Insertar equipo) Editar cada equipo (ver la sección 3 Editar equipo) Eliminar equipo (ver la sección 4 Eliminar equipo) Detalles del equipo (ver la sección 5 Detalles equipo) Exportar estadísticas a PDF (ver la sección 6 Exportar estadísticas a PDF)
Sección 2 Insertar equipo		
Flujo básico Insertar equipo		
Actor		**Sistema**
1	Selecciona la opción que permite insertar un nuevo equipo	
2		Muestra la interfaz Insertar equipo que permite introducir los valores referentes al equipo a insertar
3	Selecciona el botón Aceptar	
4		Valida los valores introducidos del equipo y muestra un mensaje de información "Equipo Insertado"
Flujo alterno Insertar equipo		
Actor		**Sistema**
1	En caso de hacer clic en el botón Aceptar y haber dejado algún campo en blanco	
2		El sistema muestra el mensaje de error “Todos los campos son obligatorios”
Sección 3 Editar equipo		
Flujo básico Editar equipo		
Actor		**Sistema**
1	Selecciona la opción que permite editar el equipo	

2		Muestra la interfaz Editar equipo que permite editar los valores referentes al equipo
3	Selecciona el botón Aceptar	
4		Valida los nuevos valores introducidos del equipo y muestra un mensaje de información "El Equipo ha sido editado"
Flujo alterno Editar equipo		
Actor		**Sistema**
1	En caso de dar clic en el botón Aceptar y haber dejado algún campo en blanco	
2		El sistema muestra el mensaje de error “Todos los campos son obligatorios”
Sección 4 Eliminar equipo		
Flujo básico Eliminar equipo		
Actor		**Sistema**
1	Selecciona la opción que permite eliminar equipo	
2		Muestra un mensaje de información: “Este equipo se eliminará”
3	Selecciona el botón Confirmar	
4		Elimina el equipo seleccionado y actualiza la lista
Flujo alterno Eliminar equipo		
Actor		
1	Una vez seleccionado el equipo presiona el botón cancelar	
2		Cancela la petición y muestra nuevamente el listado de los equipos.
Sección 5 Detalles equipo		
Flujo básico Detalles equipo		
Actor		**Sistema**
1	Selecciona uno de los equipos listados	

2		Se muestra una interfaz con la información referente al equipo seleccionado.
Sección 5 Exportar estadísticas a PDF		
Flujo básico Exportar estadísticas a PDF		
Actor		**Sistema**
1	Una vez seleccionado un equipo, presiona el botón exportar	
2		Exporta las estadísticas del equipo seleccionado a PDF
Relaciones	**CU incluidos**	**-**
Prototipo elemental de interfaz gráfica de usuario		




86% 9:00 PM
Nuevo Equipo
Nombre:

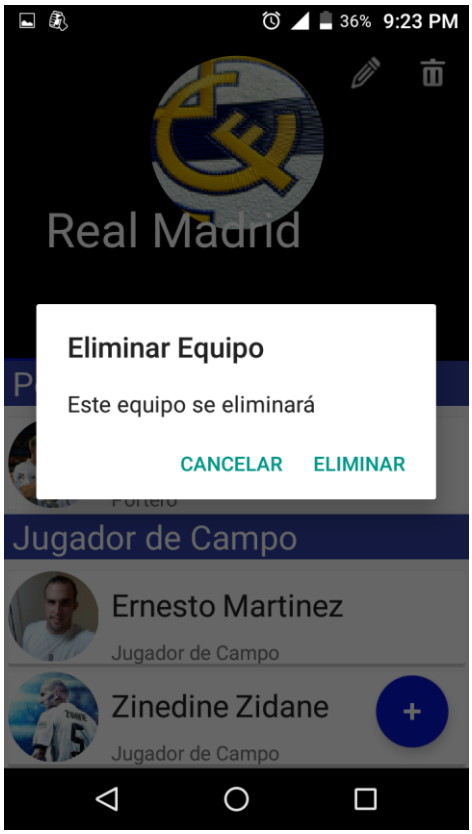
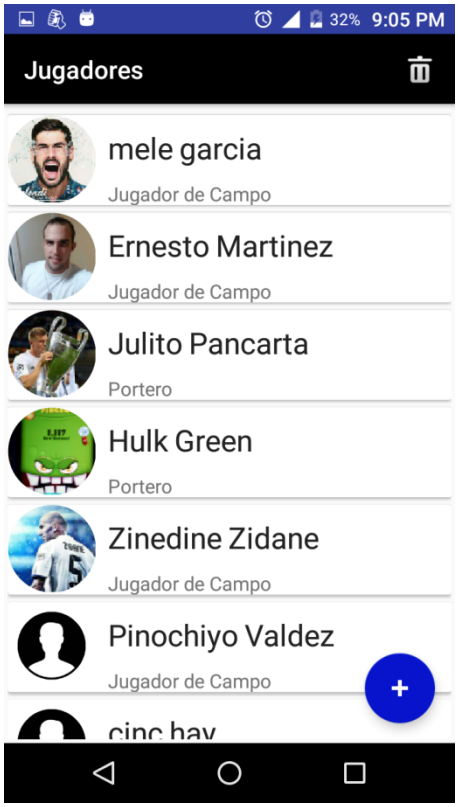

36% 9:23 PM
Real Madrid
Eliminar Equipo
Este equipo se eliminará
CANCELAR
ELIMINAR
Jugador de Campo
Ernesto Martinez
Jugador de Campo
Zinedine Zidane
Jugador de Campo

Tabla 10: CUS Gestionar jugadores

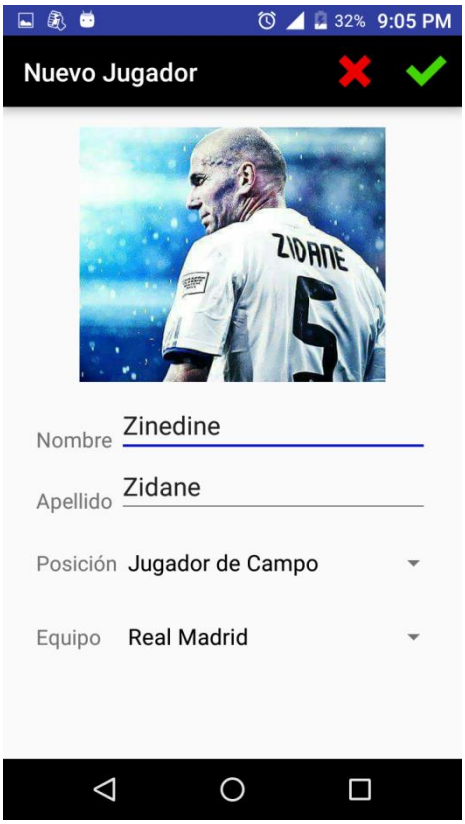
Objetivo	Gestionar los jugadores
Actores	Usuario
Resumen	El caso de uso inicia cuando el usuario desea insertar, listar, editar, ver detalles o eliminar jugadores
Complejidad	Media
Prioridad	Media
Precondiciones	El usuario selecciona la opción Jugadores
Postcondiciones	Se inserta, lista, edita o elimina los jugadores
Flujo de eventos	
Flujo básico Gestionar jugadores	
Actor	**Sistema**
Indica que desea insertar, listar, editar, ver detalles o eliminar un jugador	Si el usuario decide listar los jugadores Ver Sección 1: Listar jugadores Si el usuario decide insertar un nuevo jugador Ver Sección 2: Insertar jugador Si el usuario decide editar algún jugador Ver Sección 3: Editar jugador Si el usuario decide eliminar jugador Ver Sección 4: Eliminar jugador Si el usuario decide ver los detalles de algún jugador Ver Sección 5: Detalles jugador Si el usuario decide exportar las estadísticas a PDF. Ver Sección 6: Exportar estadísticas a PDF
Sección 1: Listar jugador	
Flujo básico Listar jugador	
Actor	**Sistema**
1 Selecciona la opción Jugadores del menú principal	
2	Muestra una interfaz con una lista de todos los jugadores existentes en la base de datos. Las opciones que se brindan son: Insertar un jugador (ver la sección 2 Insertar Jugador)

		Editar cada jugador (ver la sección 3 Editar Jugador) Eliminar jugador (ver la sección 4 Eliminar Jugador) Detalles del jugador (ver la sección 5 Detalles Jugador) Exportar estadísticas a PDF (ver la sección 6 Exportar estadísticas a PDF)
Sección 2 Insertar jugador		
Flujo básico Insertar jugador		
Actor		**Sistema**
1	Selecciona la opción que permite insertar un nuevo jugador	
2		Muestra la interfaz Insertar jugador que permite introducir los valores referentes al jugador a insertar
3	Selecciona el botón Aceptar	
4		Valida los valores introducidos del jugador y muestra un mensaje de información “Jugador Insertado"
Flujo Alterno Insertar jugador		
Actor		**Sistema**
1	En caso de hacer clic en el botón Aceptar y haber dejado algún campo en blanco	
2		El sistema muestra el mensaje de error “Todos los campos son obligatorios”
Sección 3 Editar jugador		
Flujo básico Editar jugador		
Actor		**Sistema**
1	Selecciona la opción que permite editar el jugador	
2		Muestra la interfaz Editar jugador que permite editar los valores referentes al jugador

3	Selecciona el botón Aceptar	
4		Valida los nuevos valores introducidos del jugador y muestra un mensaje de información "El Jugador ha sido editado"
Flujo alterno Editar jugador		
Actor		**Sistema**
1	En caso de dar clic en el botón Aceptar y haber dejado algún campo en blanco	
2		El sistema muestra el mensaje de error “Todos los campos son obligatorios”
Sección 4 Eliminar jugador		
Flujo básico Eliminar jugador		
Actor		**Sistema**
1	Selecciona la opción que permite eliminar jugador	
2		Muestra un mensaje de información: “Este jugador se eliminará”
3	Selecciona el botón Confirmar	
4		Elimina el jugador seleccionado y actualiza la lista
Flujo alterno Eliminar jugador		
Actor		
1	Una vez seleccionado el jugador presiona el botón cancelar	
2		Cancela la petición y muestra nuevamente el listado de los jugadores.
Sección 5 Detalles jugador		
Flujo básico Detalles jugador		
Actor		**Sistema**
1	Hace clic sobre uno de los jugadores listados	
2		Se muestra una interfaz con la información referente al jugador seleccionado.

<table>
<tr><td colspan="4">Sección 6 Exportar estadísticas a PDF</td></tr>
<tr><td colspan="4">Flujo básico Exportar estadísticas a PDF</td></tr>
<tr><td colspan="2">Actor</td><td colspan="2">Sistema</td></tr>
<tr><td>1</td><td colspan="2">Una vez seleccionado un jugador, presiona el botón exportar</td><td></td></tr>
<tr><td>2</td><td colspan="2"></td><td>Exporta las estadísticas del jugador seleccionado a PDF</td></tr>
<tr><td>Relaciones</td><td>CU incluidos</td><td colspan="2">-</td></tr>
<tr><td colspan="4">Prototipo elemental de interfaz gráfica de usuario</td></tr>
<tr><td colspan="4"> 
</td></tr>
</table>


38% 9:28 PM
Zinedine Zidane
Equipo: Real Madrid
Posicion: Jugador de Campo
Goles/Disparos: 18/19
Efectividad Total: 94%
Goles de 6m: 8 -> Efectividad: 88%
Goles de 7m: 8 -> Efectividad: 100%
Goles de 9m: 2 -> Efectividad: 100%
Asistencias: 5
Perdidas de balón: 2
Recuperaciones: 3
Tarjetas Amarillas: 5


32% 9:05 PM
Nuevo Jugador
Nombre Zinedine
Apellido Zidane
Posición Jugador de Campo
Equipo Real Madrid

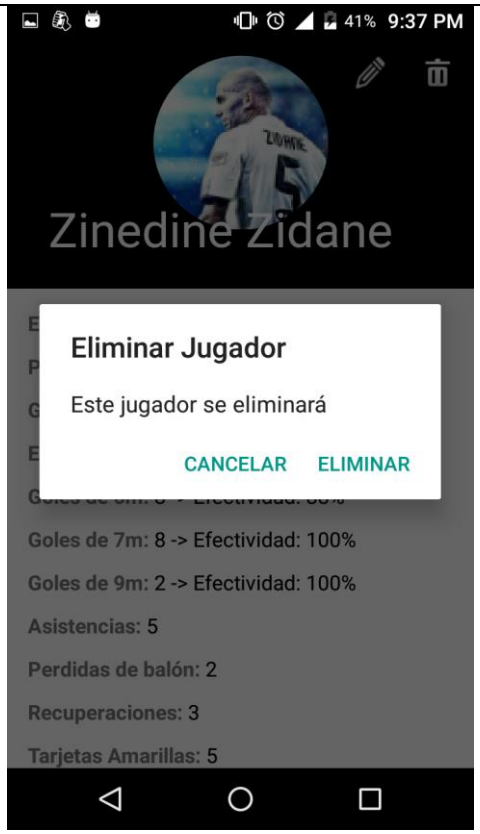
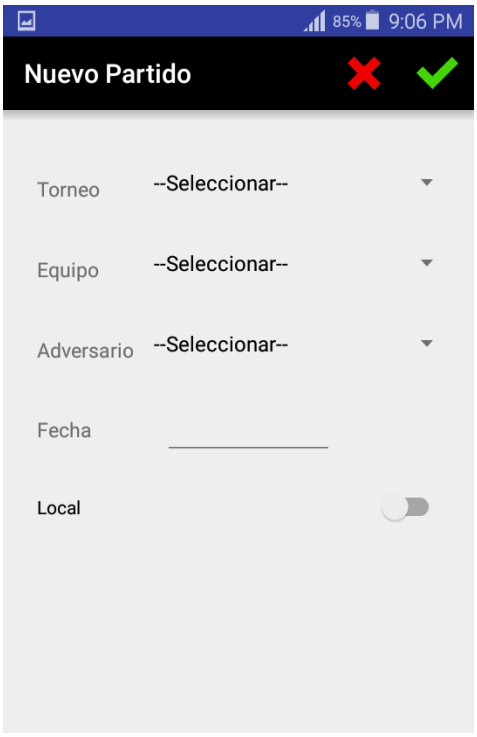


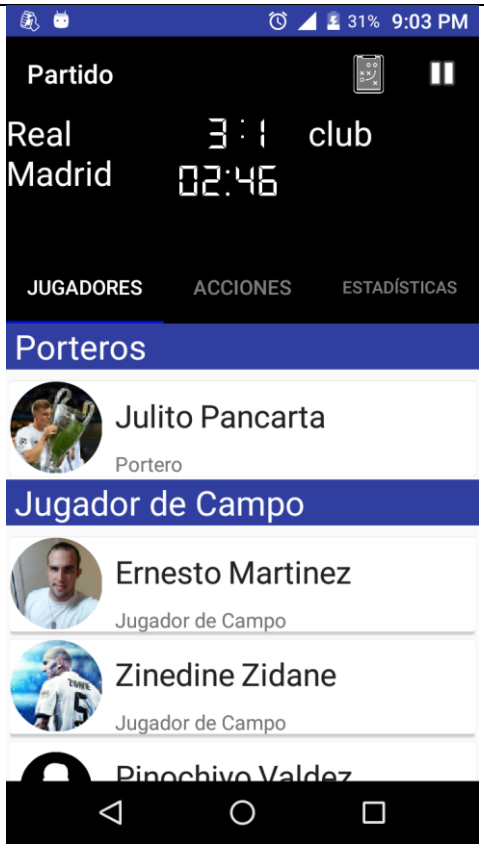
Tabla 11: CUS Crear nuevo partido

Objetivo	Registrar un partido
Actores	Usuario
Resumen	El caso de uso inicia cuando el usuario desea iniciar un nuevo partido
Complejidad	Media
Prioridad	Media
Precondiciones	El usuario selecciona la opción Nuevo Partido
Postcondiciones	Se inserta y registran las acciones de un partido
Flujo de eventos	
Flujo básico Iniciar Partido	
Actor	**Sistema**
Indica que desea iniciar un nuevo partido	Si el usuario decide iniciar un nuevo partido. Ver Sección 1: Crear nuevo partido. Si el usuario decide insertar acción del partido. Ver Sección 2: Insertar acción. Si el usuario decide ver las acciones del

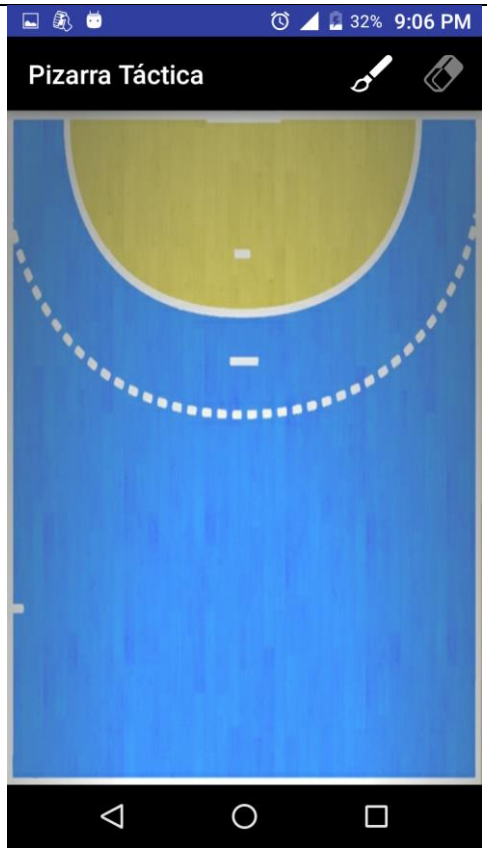
		partido. Ver Sección 3: Registro de acciones del partido. Si el usuario decide ver las estadísticas del partido. Ver Sección 4: Estadísticas.
Sección 1: Crear nuevo partido		
Flujo básico Crear nuevo partido		
Actor		**Sistema**
1	Selecciona la opción Nuevo Partido del menú principal	
2		Muestra la interfaz Insertar Equipo que permite introducir los valores referentes al partido a insertar
3	Selecciona el botón Aceptar	
		Valida los valores introducidos del partido y muestra una interfaz con el listado de los jugadores del equipo, la pizarra del partido, que incluye marcador de goles, cronómetro del partido, cronómetro de amonestaciones y los quipos que se enfrentan. Las opciones que se brindan son: Insertar acción (ver sección 2: Insertar Acción.) Registro de acciones (ver Sección 3: Registro de Acciones del Partido.) Estadísticas del partido (ver Sección 4: Estadísticas.)
Sección 2 Insertar acción		
Flujo básico Insertar acción		
Actor		**Sistema**
1	Selecciona el jugador que realizó la acción a insertar.	
2		Inserta la acción y la muestra en el registro de acciones
Sección 3 Registro de acciones del partido		
Flujo básico Registro de acciones del partido		
Actor		**Sistema**

1	Selecciona la opción acciones	
2		Muestra el listado de todas las acciones insertadas (ver Sección 2: Insertar Acción) del partido.
Sección 4 Estadísticas		
Flujo básico Estadísticas		
Actor		**Sistema**
1	Selecciona la opción estadísticas	
2		Muestras las estadísticas del partido calculadas a partir de las acciones insertadas (ver Sección 2: Insertar Acción)
Relaciones	**CU incluidos**	**-**
Prototipo elemental de interfaz gráfica de usuario		




31% 9:03 PM
Partido
Real Madrid
3 : 1
club
02:46
JUGADORES
ACCIONES
ESTADÍSTICAS
Porteros
Julito Pancarta
Portero
Jugador de Campo
Ernesto Martinez
Jugador de Campo
Zinedine Zidane
Jugador de Campo


31% 9:03 PM
Partido
Real Madrid
3 : 1
club
02:59
JUGADORES
ACCIONES
ESTADÍSTICAS
Zinedine Zidane
1' 2 minutos
0 : 1
Ernesto Martinez
2' Tarjeta roja
0 : 1
Zinedine Zidane
2' Gol 7m
1 : 1
Ernesto Martinez
2' Balón recuperado
1 : 1
Zinedine Zidane
2' Asistencia
1 : 1


32% 9:06 PM
Pizarra Táctica

Anexo 2: Diseño de casos de prueba

Tabla 12: Caso de prueba Insertar equipo

Escenario	Descripción	Nombre	Logo	Respuesta del sistema	Flujo central
EC 3.1 Insertar equipo	Permite al usuario crear un nuevo equipo.	V	V	1- Inserta en la base de datos el equipo creado por el usuario, para ello debe llenar los campos • Nombre • Logo 2- Muestra el mensaje “Equipo Insertado”	1- Seleccionar del menú principal la opción "Equipos" 2- Selecciona el botón Agregar ubicado en la esquina inferior derecha 3- Introduce los datos del equipo 4- Selecciona el botón Aceptar
		V	V vacío		
EC 3.2 Insertar equipo	Validar que no existen campos vacíos	I vacío	V	1- En el caso en que el campo Nombre esté vacío muestra el mensaje “Debe introducir el nombre del equipo”	1- Seleccionar del menú principal la opción " Equipos" 2- Selecciona el botón Agregar ubicado en la esquina inferior derecha 3- Introduce los datos del Equipo 4- Selecciona el botón Aceptar
		V	V vacío		
EC 3.3 Insertar equipo	Validar que no existen campos inválidos	I F@cultad	V	1- No permite introducir caracteres inválidos	1- Seleccionar del menú principal la opción " Equipos" 2- Selecciona el botón Agregar ubicado en la esquina inferior derecha 3- Introduce los datos del Equipo 4- Selecciona el botón Aceptar
		V	V		

Tabla 13: Caso de prueba Insertar jugador

Escenario	Descripción	Nombre	Apellido	Posición	Equipo	Foto	Respuesta del sistema	Flujo central
EC 4.1 Insertar jugador	Permite al usuario crear un nuevo jugador.	V	V	V	V	V	1- Inserta en la base de datos el jugador creado por el usuario, para ello debe llenar los campos • Nombre • Apellido • Posición • Equipo 2- Muestra el mensaje “Jugador Insertado”	5- Seleccionar del menú principal la opción "Jugadores" 6- Selecciona el botón Agregar ubicado en la esquina inferior derecha 7- Introduce los datos del jugador 8- Selecciona el botón Aceptar
		V	V	V	V vacío	V vacío		
EC 4.2 Insertar jugador	Validar que no existen campos vacíos	I vacío	V	V	V	V	1- En el caso en que los campos Nombre y Apellido estén vacíos muestra el mensaje “Todos los campos son obligatorios”	5- Seleccionar del menú principal la opción "Jugadores" 6- Selecciona el botón Agregar ubicado en la esquina inferior derecha 7- Introduce los datos del jugador 8- Selecciona el botón Aceptar
		V	I vacío	V	V	V		
		V	V	V	V	V		
		V	V	V	V vacío	V		
		V	V	V	V	V vacío		

EC 4.3 Insertar jugador	Validar que no existen campos inválidos	I Fr@nk	V	V	V	V	1- No permite introducir caracteres inválidos	5- Seleccionar del menú principal la opción "Jugadores" 6- Selecciona el botón Agregar ubicado en la esquina inferior derecha 7- Introduce los datos del jugador 8- Selecciona el botón Aceptar
		V	I G4rc1A	V	V	V		
		V	V	V	V	V		
		V	V	V	V vacío	V		
		V	V	V	V	V vacío		

Tabla 14: Caso de prueba Insertar torneo

Escenario	Descripción	Nombre	Cantidad de Equipos	Tipo de Torneo	Duración de Partidos	Fecha de Inicio	Fecha de Fin	Respuesta del sistema	Flujo central
EC 5.1 Insertar torneo	Permite al usuario crear un nuevo torneo.	V	V	V	V	V	V	1- Inserta en la base de datos el torneo creado por el usuario, para ello debe llenar los campos • Nombre • Cantidad de Equipos • Tipo de Torneo • Duración de Partidos • Fecha de Inicio • Fecha de Fin 2- Muestra el mensaje “Torneo Insertado”	1- Seleccionar del menú principal la opción "Torneos" 2- Selecciona el botón Agregar ubicado en la esquina inferior derecha 3- Introduce los datos del torneo 4- Selecciona el botón Aceptar
EC 5.2 Insertar torneo	Validar que no existen campos vacíos	I vacío	V	V	V	V	V	1- Muestra el mensaje “Todos los campos son obligatorios”	1- Seleccionar del menú principal la opción "Torneos" 2- Selecciona el botón Agregar ubicado en la esquina inferior derecha
		V	I vacío	V	V	V	V		
		V	V	V	V	V	V		

		V	V	V	I vacío	V	V		3- Introduce los datos del torneo 4- Selecciona el botón Aceptar
		V	V	V	V	I vacío	V		
		V	V	V	V	V	I vacío		
EC 5.3 Insertar torneo	Validar que no existen campos inválidos	I Ju3go$	V	V	V	V	V	1- No permite introducir estos caracteres. 2- Muestra el mensaje “Para los torneos por grupos la cantidad de equipos debe ser par” 3- Muestra el mensaje “La duración mínima de un partido es de 2 minutos” 4- Muestra el mensaje “La fecha de fin debe ser mayor que la fecha de inicio	1- Seleccionar del menú principal la opción "Torneos" 2- Selecciona el botón Agregar ubicado en la esquina inferior derecha 3- Introduce los datos del torneo 4- Selecciona el botón Aceptar
		V	I 9	V	V	V	V		
		V	V	I Por grupos	V	V	V		
		V	V	V	I 1	V	V		
		V	V	V	V	I 26/04/17	V		
		V	V	V	V	V	I 20/04/17		