



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
CENTRO DE INVESTIGACIÓN EN COMUNICACIÓN Y REDES

Desarrollo de una solución tecnológica para la recolección, almacenamiento y consulta de datos durante un combate de Judo utilizando tecnologías móviles

**Trabajo Especial de Grado presentado ante la ilustre
Universidad Central de Venezuela por el
Br. Humberto José Rodrigues García
Br. Javier Antonio Flores Vera**

**Tutor:
Prof. Ana Verónica Morales Bezeira
Abril de 2015**

Universidad Central de Venezuela.
Facultad de Ciencias
Escuela de Computación
Centro de Investigación en Comunicación y Redes

Desarrollo de una solución tecnológica para la recolección, almacenamiento y consulta de datos durante un combate de Judo utilizando tecnologías móviles

Autores: Humberto José Rodrigues García

Javier Antonio Flores Vera

Tutora: Profa. Ana Verónica Morales Bezeira

Fecha: 14 de Abril de 2015

Palabras Claves: recolección, almacenamiento, consulta, desarrollo, deporte, Java, Judo, Android, Móvil, Escritorio, exportación, datos.

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro de Investigación en Comunicación y Redes (CICORE)

ACTA

Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación para examinar el Trabajo Especial de Grado, presentado por los Bachilleres Javier Antonio Flores Vera C.I.:20.595.080 y Humberto José Rodrigues García C.I.: 22.727.526, titulado "**Desarrollo de una solución tecnológica para la recolección, almacenamiento y consulta de datos durante un combate de Judo utilizando tecnologías móviles**", a los fines de cumplir con el requisito legal para optar al título de Licenciada en Computación, dejan constancia de lo siguiente:

Leído el trabajo por cada uno de los miembros del jurado, se fijó el día 10 de Abril de 2015, a las 09:00 am, para que sus autores lo defendieran de forma pública, en la sala de videoconferencias del Centro de Computación de la Facultad de Ciencias, lo cual estos realizaron mediante una exposición oral de su contenido, y luego respondieron satisfactoriamente a las preguntas que le fueron formuladas por el jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fe de lo cual se levanta la presente acta, en Caracas el 14 de Abril de 2015, dejándose también constancia de que actuó como Coordinador del jurado la Profesora Tutora Ana Verónica Morales Bezeira.

Prof. Ana Morales
(Tutor)

Prof. Jesús Lares
(Jurado Principal)

Prof. Jesús Gómez
(Jurado Principal)

AGRADECIMIENTOS

A nuestros padres, por apoyarnos en todo momento.

A la profesora María Elena Villapol Blanco, por ser protagonista en este trabajo de investigación, agregando las primeras ideas al proyecto. Además, por estar siempre disponible para responder nuestras dudas y ayudarnos en diversas áreas, aún en las situaciones más adversas.

A nuestra tutora, Ana Verónica Morales Bezeira, por brindarnos su ayuda, su tiempo y su solidaridad en relación a los problemas metodológicos, técnicos y teóricos que se nos presentaron durante el desarrollo de la solución de *Software*, consiguiendo tiempo para reuniones, revisiones y lecturas incluso en los momentos más difíciles.

Al Sensei Hernan Jansen, por ayudarnos en los aspectos teóricos y técnicos del Judo, así como agregar numerosas ideas para enriquecer el sistema.

Al Club de Judo de la UCV, por permitirnos entrar a sus instalaciones a realizar pruebas e intervenir durante sus entrenamientos y competencias. Extensivo a todos los miembros que colaboraron durante prácticas y competencias en las cuales se realizaron pruebas del sistema.

A todos los que con sus ideas, sugerencias y colaboración apoyaron este trabajo de investigación.

A CICORE, por darnos la oportunidad de aprender e investigar en sus instalaciones.

ÍNDICE GENERAL

ÍNDICE GENERAL.....	I
ÍNDICE DE FIGURAS	V
ACRÓNIMOS	VII
RESUMEN	VIII
INTRODUCCIÓN	IX
CAPÍTULO 1: EL PROBLEMA.....	10
1.1 PLANTEAMIENTO DEL PROBLEMA	10
1.2 OBJETIVO GENERAL	11
1.3 OBJETIVOS ESPECÍFICOS	11
1.4 JUSTIFICACIÓN.....	11
1.5 ANTECEDENTES	12
1.6 ALCANCE.....	13
CAPÍTULO 2: MARCO CONCEPTUAL	15
2.1 ¿QUÉ ES ANDRIOD?	15
2.1.1 FUNCIONAMIENTO	15
2.1.2 APLICACIONES.....	16
2.2 MECANISMOS DE EXPORTACION DE DATOS DESDE UN DISPOSITIVO ANDROID	16
2.2.1 EXPORTACIÓN UTILIZANDO ARCHIVOS DE FORMATO CONOCIDO.....	16
2.2.2 EXPORTACIÓN UTILIZANDO HERRAMIENTAS NATIVAS DE ANDROID.....	16
2.2.3 EXPORTACIÓN UTILIZANDO INTERNET	17
2.3 MECANISMOS DE CAPTURA Y MANIPULACIÓN DE LA CAMARA EN ANDROID.....	17
2.4 MECANISMOS DE TEMPORIZACIÓN EN DISPOSITIVOS ANDROID.....	18
2.5 CONEXIÓN A BASE DE DATOS DESDE ANDROID.....	19
2.6 HERRAMIENTAS DE DESARROLLO	19
2.6.1 LENGUAJE DE PROGRAMACIÓN JAVA	19
2.6.2 LENGUAJE DE PROGRAMACIÓN PHP.....	21
2.6.3 HERRAMIENTAS DE DESARROLLO EN ANDROID	22
2.6.4 API DE GOOGLE DRIVE	25
2.6.5 JSON	25
2.6.6 GSON.....	26

2.6.7 BIBLIOTECA CHARTS4J	26
2.6.8 SUGAR ORM.....	27
2.6.9 SERVIDOR HTTP APACHE	28
2.6.10 SHA-256	28
2.6.11 MP4PARSER	29
2.6.12 HERRAMIENTAS SELECCIONADAS	29
2.7 MECANISMOS DE ALMACENAMIENTO	31
2.7.1 SQLITE	31
2.7.2 MYSQL.....	31
2.7.3 GOOGLE DRIVE.....	32
2.8 EVENTOS Y REGLAS DE LOS COMBATES DE JUDO	33
2.8.1 REGLAS DEL JUDO	33
2.8.2 EVENTOS EN LOS COMBATES DE JUDO	35
CAPÍTULO 3: MÉTODO DE DESARROLLO	39
3.1 MODELO DE PROCESOS DE INGENIERÍA DE SOFTWARE	39
3.1.1 MODELO DE DESARROLLO ÁGIL	39
3.1.2 MODELO BASADO EN PROTOTIPOS.....	40
3.2 MÉTODO DE DESARROLLO SELECCIONADO	41
CAPÍTULO 4: DESARROLLO DE LA SOLUCIÓN	43
4.1 ARQUITECTURA DE LA SOLUCIÓN	43
4.2 ANÁLISIS Y DISEÑO DEL SISTEMA	44
4.3 DESARROLLO	71
4.3.1 DESARROLLO DE APLICACIÓN MÓVIL.....	71
4.3.1.1 PROTOTIPO INICIAL.....	71
4.3.1.2 REVISIONES DEL PROTOTIPO INICIAL POR PARTE DEL CLIENTE	72
4.3.1.3 SEGUNDO PROTOTIPO	73
4.3.1.4 REVISIONES DEL SEGUNDO PROTOTIPO POR PARTE DEL CLIENTE	74
4.3.1.5 TERCER PROTOTIPO	74
4.3.1.6 REVISIONES DEL TERCER PROTOTIPO POR PARTE DEL CLIENTE.....	75
4.3.1.7 PROTOTIPO FINAL.....	75
4.3.2 DESARROLLO DE APLICACIÓN DE ESCRITORIO	76
4.3.2.1 PROTOTIPO INICIAL.....	76
4.3.2.2 REVISIONES DEL PROTOTIPO INICIAL	86

4.3.2.3 PROTOTIPO FINAL.....	87
4.3.3 DESARROLLO DE INTERFAZ DE COMUNICACIÓN	88
4.4 PRUEBAS.....	88
4.5 IMPLEMENTACIÓN	90
CAPÍTULO 5: CONCLUSIONES	92
5.1 CONTRIBUCIÓN	92
5.2 RECOMENDACIONES	92
5.3 LÍMITES	93
5.4 TRABAJOS FUTUROS.....	94
REFERENCIAS	95
ANEXOS.....	98

ÍNDICE DE FIGURAS

ILUSTRACIÓN 1 CONEXIÓN A BASE DE DATOS SQLITE DESDE JAVA	19
ILUSTRACIÓN 2 CONEXIÓN A BASE DE DATOS MYSQL DESDE JAVA	21
ILUSTRACIÓN 3 CONEXIÓN A BASE DE DATOS MYSQL DESDE PHP.....	22
ILUSTRACIÓN 4 ESTRUCTURA DE UN OBJETO JSON	26
ILUSTRACIÓN 5 EJEMPLO DE USO DE LA BIBLIOTECA GSON	26
ILUSTRACIÓN 6 SUBCONJUNTO DE CÓDIGO DE GRAFICO UTILIZANDO CHARTS4J API	27
ILUSTRACIÓN 7 BÚSQUEDA Y MODIFICACIÓN DE UN OBJETO DEL MODELO DE DATOS UTILIZANDO SUGAR ORM	28
ILUSTRACIÓN 8 MAQUINA DE ESTADO DE LA CLASE MEDIARECORDER [26]	30
ILUSTRACIÓN 9 ÁREA DE COMPETENCIA DEL JUDO [30].....	33
ILUSTRACIÓN 10 IPPON EN EJECUCIÓN [1]	36
ILUSTRACIÓN 11 WAZA-ARI POR INMOVILIZACIÓN EN EJECUCIÓN	37
ILUSTRACIÓN 12 . TABLA DE PUNTUACIÓN.....	38
ILUSTRACIÓN 13 FASES DEL MÉTODO DE DESARROLLO	41
ILUSTRACIÓN 14 ARQUITECTURA DE LA SOLUCIÓN DE SOFTWARE	43
ILUSTRACIÓN 15 DIAGRAMA DE CASOS DE USO NIVEL 1 DE APLICACIÓN MÓVIL	45
ILUSTRACIÓN 16 DIAGRAMA DE CASOS DE USO NIVEL 2 DE APLICACIÓN MÓVIL	46
ILUSTRACIÓN 17 DIAGRAMA DE CASOS DE USO NIVEL 1 DE APLICACIÓN DE ESCRITORIO	57
ILUSTRACIÓN 18 DIAGRAMA DE CASOS DE USO NIVEL 2 DE APLICACIÓN DE ESCRITORIO	58
ILUSTRACIÓN 19 DIAGRAMA DE CLASES DEL SISTEMA	68
ILUSTRACIÓN 20 MODELO DE BASE DE DATOS SQLITE DE APLICACIÓN MÓVIL	69
ILUSTRACIÓN 21 MODELO DE BASE DE DATOS MYSQL DE APLICACIÓN DE ESCRITORIO	70
ILUSTRACIÓN 22 INTERFAZ DE GRABACIÓN DE COMBATE DEL PROTOTIPO INICIAL.....	72
ILUSTRACIÓN 23 INTERFAZ DE GRABACIÓN DE COMBATE DEL SEGUNDO PROTOTIPO	73
ILUSTRACIÓN 24 INTERFAZ DE GRABACIÓN DE COMBATE DEL TERCER PROTOTIPO	75
ILUSTRACIÓN 25 INTERFAZ DE COMBATE DEL PROTOTIPO FINAL DE LA APLICACIÓN MÓVIL	76
ILUSTRACIÓN 26 INTERFAZ PRINCIPAL DEL PROTOTIPO INICIAL DE LA APLICACIÓN DE ESCRITORIO	77
ILUSTRACIÓN 27 GRÁFICA DE MARCACIONES	78
ILUSTRACIÓN 28 GRÁFICA DE COMBATES	79
ILUSTRACIÓN 29 GRÁFICA DE FRECUENCIA DE ATAQUE.....	80
ILUSTRACIÓN 30 GRÁFICA DE PERFIL GENERAL	81
ILUSTRACIÓN 31 GRÁFICA DE ESTILO DE MARCACIÓN	82
ILUSTRACIÓN 32 GRÁFICA DE COMBATES DE GOLDEN SCORE	83
ILUSTRACIÓN 33 GRÁFICA DE COMBATES EN TIEMPO REGLAMENTARIO	83
ILUSTRACIÓN 34 GRÁFICA DE FRECUENCIA DE ATAQUE (MARCACIONES) DE UN COMBATE	84
ILUSTRACIÓN 35 GRÁFICA DE FRECUENCIA DE ATAQUE (PENALIZACIONES) DE UN COMBATE	85
ILUSTRACIÓN 36 GRÁFICA DE ESTILO DE MARCACIÓN	86
ILUSTRACIÓN 37 INTERFAZ PRINCIPAL DEL PROTOTIPO FINAL DE LA APLICACIÓN DE ESCRITORIO.....	87
ILUSTRACIÓN 38 RESPUESTAS DE LA PREGUNTA: ¿CONSIDERA USTED QUE LA APLICACIÓN MÓVIL ES FÁCIL DE USAR? EN LA PRUEBA DE ACEPTACIÓN	88
ILUSTRACIÓN 39 RESPUESTAS DE LA PREGUNTA: ¿CONSIDERA USTED QUE LA APLICACIÓN MÓVIL PERMITE CAPTURAR LOS EVENTOS OCURRIDOS EN EL COMBATE Y SU VÍDEO ASOCIADO? EN LA PRUEBA DE ACEPTACIÓN.....	89
ILUSTRACIÓN 40 RESPUESTAS DE LA PREGUNTA: ¿CONSIDERA USTED QUE LA APLICACIÓN MÓVIL AYUDA A AGILIZAR EL PROCESO DE RECOLECCIÓN DE LOS DATOS DE EVENTOS OCURRIDOS EN UN COMBATE DE JUDO? EN LA PRUEBA DE ACEPTACIÓN	89
ILUSTRACIÓN 41 RESPUESTAS DE LA PREGUNTA: ¿CONSIDERA USTED QUE LOS DATOS QUE MUESTRA LA APLICACIÓN DE ESCRITORIO RELACIONADOS A LOS COMBATES SON CORRECTOS Y SE VISUALIZAN DE UNA FORMA ADECUADA? EN LA PRUEBA DE ACEPTACIÓN	89

ILUSTRACIÓN 42 RESPUESTAS DE LA PREGUNTA: ¿ESTÁ USTED DE ACUERDO EN QUE LOS DATOS DE LOS COMBATES CAPTURADOS CON LA APLICACIÓN MÓVIL PUEDEN VISUALIZARSE DESDE LA APLICACIÓN DE ESCRITORIO? EN LA PRUEBA DE ACEPTACIÓN	90
ILUSTRACIÓN 43 RESPUESTAS DE LA PREGUNTA: ¿CREE USTED QUE LA APLICACIÓN DE ESCRITORIO MEJORA LAS HERRAMIENTAS QUE AYUDAN AL ANÁLISIS DE LOS COMBATES DE JUDO? EN LA PRUEBA DE ACEPTACIÓN	90

ACRÓNIMOS

IDE	Integrated Development Environment
USB	Universal Serial Bus
ADT	Android Development Tools
Android SDK	Android <i>Software</i> Development Kit
API	Application Programming Interface
WiFi	Mecanismo de conexión inalámbrica basado en el conjunto de estándares 802.11
JSON	JavaScript Object Notation
UNICODE	Estándar: Universalidad, uniformidad y unicidad.
HTTP	Hypertext Transfer Protocol
PHP	Hypertext Preprocessor
URL	Uniform Resource Locator
GB	Gigabyte
NIST	National Institute of Standards and Technology
ORM	Object-Relational mapping
APK	Application Package File
JAR	Java Archive

RESUMEN

La recolección y análisis de datos es un área en expansión ampliamente utilizada en los últimos años. Los deportes profesionales no escapan de este hecho, ya que cada disciplina genera una gran cantidad de datos.

Los mecanismos utilizados para la recolección de datos generalmente requieren que sean adaptados a cada disciplina, debido a que dependen de las características de ésta. Actualmente la Selección Venezolana de Judo no cuenta con un mecanismo eficaz para recolectar datos de forma eficiente causando así pérdida o mala interpretación de estos, y por consiguiente, dificulta la planificación del entrenamiento de sus atletas.

Motivado a esto, la siguiente investigación tiene como fin desarrollar una solución tecnológica para la recolección, almacenamiento y consulta de los datos que surgen durante un combate de Judo, utilizando como plataforma base un dispositivo móvil, específicamente con tecnología *Android*. La aplicación móvil cuenta con una interfaz de usuario basada en mecanismos de interacción táctil y botones asociados a diferentes eventos característicos del Judo. Gracias a esto, se minimiza la pérdida de información. Otra característica importante es que la solución permite exportar los datos recolectados a un computador, con el fin de analizarlos libremente con las herramientas disponibles.

INTRODUCCIÓN

Entre las disciplinas deportivas se utilizan mecanismos para capturar datos de los deportistas. Frecuentemente se realiza mediante una videocámara. La disciplina de Judo tiene la característica que durante el desenvolvimiento de sus combates surge, con rapidez, una gran cantidad de información. Para el momento de realizar este trabajo de investigación, la selección nacional de Judo Venezolana no posee los medios ni las herramientas adecuadas para recolectar dichos datos de forma eficiente causando posibles pérdidas o mala interpretación de ellos, y esto impacta directamente en el entrenamiento de sus atletas.

De allí la importancia de la realización del presente trabajo de investigación, el cual desarrolla una solución tecnológica que permita la recolección, almacenamiento y consulta de los datos generados durante un combate de Judo, de forma tal que facilite la toma de decisiones, utilizando Tecnologías de Información y Comunicación móvil.

Este documento consta de los siguientes capítulos:

El Capítulo 1 explica el problema planteado, los objetivos, justificación, antecedentes y alcance de esta investigación.

El Capítulo 2 contiene el marco teórico, en el cual se realiza una breve descripción del sistema operativo *Android*, sus aplicaciones y funcionalidades básicas que permiten desarrollar aplicaciones en dicho sistema. Describe las herramientas utilizadas para el desarrollo del *Software*, junto con sus características principales. Además, lista y describe brevemente los eventos y reglas del Judo dictados por la Federación Internacional de Judo.

El Capítulo 3 evidencia el método de desarrollo de *Software* seleccionado para utilizar en este trabajo de investigación.

El Capítulo 4 explica la arquitectura y detalles de la solución propuesta para la recolección, almacenamiento y consulta de los datos generados por los combates de Judo. Se muestra en detalle la descripción, funcionamiento y elaboración de las herramientas desarrolladas.

El Capítulo 5 muestra las conclusiones, limitaciones, recomendaciones y futuros trabajos que pueden ser realizados utilizando las herramientas generadas en esta investigación.

CAPÍTULO 1: EL PROBLEMA

En este capítulo se plantea el problema en relación a la captura, almacenamiento y visualización de los datos generados durante un combate de Judo, el impacto que conlleva esto en el entrenamiento de los judoca, se señala el objetivo general del presente trabajo de investigación y los objetivos específicos para lograrlo, además se justifica su realización basado en el impacto que tendría en contramedida con los requerimientos necesarios para su realización, se establecen antecedentes de investigaciones anteriores a este trabajo y se delimita su alcance.

1.1 PLANTEAMIENTO DEL PROBLEMA

Durante las competencias deportivas se genera una gran cantidad de datos los cuales pueden ser recolectados y transformados en información. Al paso del tiempo y en muchas disciplinas deportivas se ha utilizado el método de captura de información mediante una videocámara para luego ser estudiado por los entrenadores de la misma. La disciplina de Judo tiene la característica de que durante el desenvolvimiento de sus combates surge, con rapidez, una gran cantidad de información.

A lo largo de un combate de Judo suceden una serie de eventos relacionados con la dinámica del mismo, los cuales son el producto de la ejecución de las diversas técnicas que conforman la disciplina [1]. La ejecución de una técnica que conlleva a una acción podría generar una marcación o una falta para alguno de los judocas. Estas faltas o marcaciones son expresados como eventos, oficialmente la Federación Internacional de Judo [1] menciona en el reglamento de la disciplina los siguientes eventos: *Ippon*, *Waza-ari*, *Juko*, *Shido* y *Hansoku-make*, estos aplican para ambos combatientes y pueden ser aplicados en diferentes estados del combate, de pie o en el suelo. El entrenador y personal técnico a cargo de la Selección de Judo registran estas acciones con la finalidad de analizarlas posteriormente o durante el combate, proporcionando una retroalimentación al judoca, una vez finalizado el combate y de forma más detallada una vez finalizada la competencia o campeonato.

La Selección Venezolana de Judo, actualmente no posee los medios ni las herramientas que permitan recolectar los datos provenientes de los combates de forma eficiente, por lo tanto, registra los eventos producidos en los combates de forma manual en una libreta/cuaderno o algunas veces en un dispositivo electrónico (por ejemplo, *Tablet* o *laptop*) usando las facilidades de edición de texto. Este procedimiento es ineficiente debido a que, el entrenador o personal técnico debe realizar la captura al mismo tiempo que está atento al combate. Por lo cual ciertas acciones importantes no llegan a ser registradas, limitando el posterior análisis de lo sucedido en el combate y por ende la retroalimentación que se le debe dar al judoca. Adicionalmente, puede ocurrir que los datos registrados no sean lo suficientemente claros por la premura con la que fueron escritos y

como consecuencia su interpretación sea errada. En general, utilizando un método de captura manual los datos pueden perderse, interpretarse de forma incorrecta o perder la integridad de los mismos, todo esto impacta directamente en el entrenamiento de los atletas.

En la actualidad existe una serie de dispositivos (teléfonos inteligentes, tabletas, entre otros) que cuentan con una serie de herramientas —cámaras, pantallas táctiles, micrófonos— que permiten registrar una serie de eventos que ocurren alrededor del usuario del equipo.

Motivado por lo escrito anteriormente, una solución tecnológica que permita la captura de los eventos ocurridos en los combates de Judo utilizando los dispositivos electrónicos móviles actuales, puede mejorar en gran medida la situación actual de la Selección Venezolana de Judo, además, permitiría mejorar la visualización de los datos, con el objetivo de dar un mejor apoyo al entrenamiento de los atletas.

1.2 OBJETIVO GENERAL

Desarrollar una solución tecnológica que permita recolectar, almacenar, visualizar y extraer la información relevante que surge durante un combate de Judo utilizando tecnologías móviles.

1.3 OBJETIVOS ESPECÍFICOS

- Estudiar y comprender las reglas que rigen un combate de Judo.
- Analizar el proceso de registro de eventos que suceden en un combate de Judo y su posterior análisis.
- Estudiar las tecnologías de desarrollo utilizadas para aplicaciones de plataforma *Android*.
- Diseñar el sistema de registro, almacenamiento y visualización de eventos que suceden en un combate de Judo utilizando tecnologías de información y comunicación.
- Implementar el sistema de registro, almacenamiento y visualización de eventos que suceden en un combate de Judo.
- Realizar pruebas de funcionalidad y aceptación del usuario.

1.4 JUSTIFICACIÓN

La situación actual de la Selección Venezolana de Judo junto con lo expuesto anteriormente, y su método de captura de información de los combates en los que intervienen sus miembros, conlleva a indagar en posibles aplicaciones de la tecnología para mejorar dicha situación y reformar los mecanismos que utilizan actualmente para la

captura, almacenamiento y visualización de los datos de combates de Judo en los cuales participa la Selección.

Aplicar tecnologías móviles para los mecanismos mencionados anteriormente impactaría directamente en la calidad de los datos, la eficacia de la captura, la eficiencia del personal de la selección y amplía la gama de información que puede ser extraída de los datos capturados.

Adicionalmente, al capturar los datos de forma digital utilizando las tecnologías móviles permite que los datos sean almacenados en repositorios de datos que pueden hacer respaldos continuos, además los datos pueden ser analizados de forma óptima, permitiendo la posibilidad de procesos de análisis previo de los datos por parte del sistema, también permite desplegar la información de forma óptima para el usuario.

Por otro lado, debido a la naturaleza de los datos, la tecnología permite fácil acceso desde diferentes locaciones, soportado por la interconectividad de Internet. Esto, agrega la característica de poder compartir información entre entrenadores de forma rápida y sencilla.

Por la naturaleza de la disciplina de Judo, la aplicación de la tecnología para la captura, almacenamiento y visualización también permite un análisis muy rápido de los datos para la toma de decisiones que impacten el desarrollo de una competencia.

De tal manera que la aplicación de captura, los servidores de almacenamiento y la aplicación de escritorio, basados en el uso de tecnologías de libre uso, las cuales no están limitadas a la compra de una licencia comercial, permiten hacer una baja inversión económica para un sistema que apoya directamente al deporte nacional.

1.5 ANTECEDENTES

Es frecuente en las disciplinas deportivas la captura de datos que surgen durante el desarrollo de sus competencias y actividades, esta información varía mucho en relación a la disciplina y sus participantes. Estas capturas generan un gran volumen de datos y resulta muy difícil manejarlos, por ello se han desarrollado técnicas que permiten la manipulación de estos para extraer información.

En la actualidad es común la utilización de técnicas de análisis de información en la comunidad deportiva, bien sea para visualizar y manipular los datos y así obtener información, como para la detección de patrones de comportamiento de los deportistas brindando herramientas de última generación para los entrenadores y directores técnicos, lo cual les permite reforzar los conocimientos, técnicas y formación impartida. Ejemplo de esto es el equipo de fútbol Atlético de Madrid que según [2], adquirió un *Software* el cual les permite crear patrones de lesiones y así predecir cuándo puedan estas ocurrir, utilizando

la recolección de datos de los jugadores como metros recorridos por juego, número de caídas, entre otros.

Sin embargo, estas herramientas de recolección de información no solo son utilizadas para medir comportamientos y patrones de los jugadores propios sino que también es muy común utilizarlas para la obtención de conocimiento del contrincante, saber que técnica utiliza y en qué momento la utiliza, dando así la posibilidad de crear estrategias especiales para cada oponente. El *Advanced Scout* es un *Software* que emplea técnicas de minería de datos que han desarrollado investigadores de IBM (*International Business Machines*) para detectar patrones estadísticos y eventos. Un equipo de la NBA (*National Basketball Association*) adquirió este *Software* que da soporte en cada partido para determinar los puntajes que debe obtener cada jugador contrario basándose en la cantidad de pases, encestes, velocidad promedio y otros datos [3].

En relación con la disciplina de Judo, investigadores de la Universidad de A Coruña realizaron un estudio de captura de datos de los combates de Judo utilizando un programa informático, dando como resultado la posibilidad de extraer la estructura temporal del combate, ver las variaciones que surgen en los lapsos acontecidos en combate y comparar los eventos del combate en relación a su duración, cantidades y promedios [4].

1.6 ALCANCE

El presente trabajo de investigación desarrollará un sistema que permita la captura, almacenamiento y visualización de los datos que se generen durante un combate de Judo. Eso contempla el diseño y desarrollo de dos aplicaciones, una utilizando la plataforma de *Android* para dispositivos móviles de tipo *Tablet*; la segunda aplicación se implementará utilizando el lenguaje de programación Java, específicamente para el sistema operativo Windows y un computador de escritorio. Adicionalmente, se diseñará e implementará un modelo de datos que permite almacenar la información de los combates, además de utilizar un servidor para almacenar los videos de los combates, finalmente se diseñará e implementará una arquitectura de sistema que permita la migración de los datos desde el dispositivo *Android* hasta el servidor de datos, y desde el servidor de datos hasta la aplicación de escritorio.

La aplicación de plataforma *Android* tiene como objetivo la captura de los datos generados en los combates de Judo, dichos datos incluyen todos los eventos y la grabación del combate, adicionalmente, la aplicación debe permitir la sincronización de los datos del combate con el servidor. Por otro lado, el dispositivo móvil debe almacenar todos los datos de los combatientes y de los combates que no han sido sincronizados, así como la posibilidad de visualizar estos datos directamente en el dispositivo.

La aplicación desarrollada bajo el lenguaje Java debe tener la capacidad de sincronizar sus datos del servidor y permitir la visualización de los mismos, tanto los eventos como el video capturado por el dispositivo *Android*, adicionalmente debe permitir la edición

de los eventos y la posibilidad de agregar eventos y comentarios. En relación a los combatientes, la aplicación debe ser capaz de desplegar la información de cada combatiente, permitir editar la información personal de cada combatiente y agregar combatientes a la base de datos.

Ambas aplicaciones deben tener una interfaz que esté acorde con los parámetros de usabilidad aplicados a cada dispositivo, para permitir que los usuarios capturen y visualicen los datos de una manera correcta.

Para el almacenamiento de los datos, se utilizarán dos servidores, uno que almacenará los datos relacionados a los eventos de los combates, combatientes, combates, información de accesos a los videos y descripción de los eventos que se pueden generar durante el combate. Todos estos datos serán almacenados utilizando un servidor de datos MySQL. El segundo servidor será para almacenar los archivos de videos de los combates capturados, esto se realizará utilizando los servidores de Drive, un servicio de Google que permite el almacenamiento de archivos para sus usuarios, el cual dispone de bibliotecas para conectarse a sus servicios de almacenamiento de archivos.

Finalmente, la arquitectura que permite la interconexión de los diferentes servicios y terminales que están presentes en este trabajo será diseñado e implementado utilizando las ventajas técnicas del protocolo Internet, tanto el dispositivo móvil como la aplicación de escritorio deben estar conectados a la red Internet para sincronizarse correctamente.

CAPÍTULO 2: MARCO CONCEPTUAL

El presente trabajo de investigación está enfocado al desarrollo de una aplicación para la plataforma *Android*, una aplicación en lenguaje de programación Java y una arquitectura para interconectarlos utilizando tecnologías móviles de comunicación, por esto, en el presente capítulo se exponen los conceptos y definiciones de los temas necesarios para el entendimiento del mismo y de cada uno de los términos involucrados en el desarrollo del trabajo.

2.1 ¿QUÉ ES ANDRIOD?

Actualmente, más del 80% de los dispositivos móviles del mundo utilizan *Android* como sistema operativo, así lo afirma el IDC [5]. Grandes empresas desarrolladoras de dispositivos móviles lo utilizan por sus características de código abierto. Otra ventaja es la frecuencia de actualización de sus funcionalidades, lo cual lo ayuda a mantenerse al día con los requerimientos de los usuarios.

Según la documentación oficial de *Android* [6], es un sistema operativo basado en Linux para dispositivos móviles, a diferencia de su competencia *IOS* y *Windows phone* pertenece a la categoría de *Software* libre por lo que su código fuente es abierto, además de esto, los dispositivos con este sistema tienen más del 50% del mercado mundial por lo que, tiene una gran comunidad de desarrolladores en todo el mundo creando aplicaciones y nuevas funcionalidades.

2.1.1 FUNCIONAMIENTO

La documentación oficial de *Android* [7], indica que el sistema operativo tiene como núcleo Linux, a través del cual maneja primitivas de bajo nivel que le permiten gestionar la pila de memoria, los procesos, la pila de comunicaciones de red, los modelos de los controladores y algunos aspectos de seguridad. Actuando así, como una capa de abstracción entre el hardware y el resto de la pila de *Software*.

Posee un conjunto muy grande de bibliotecas de C/C++ que funcionan principalmente para ciertas rutinas básicas del sistema. Algunas de estas bibliotecas son: *System C library*, que es una implementación de la biblioteca C estándar, bibliotecas de gráficos 3D, SQLite, entre otras.

Adicionalmente, contiene un conjunto de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Brindándole así una de sus características principales, cada aplicación *Android* corre su propio proceso, utilizando su propia instancia de la máquina virtual *Dalvik*, la cual está escrita para correr procesos de manera muy eficiente y ejecuta archivos en el formato *Dalvik Executable* (.dex), optimizando la memoria utilizada.

2.1.2 APLICACIONES

Las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.

Una gran ventaja de desarrollar aplicaciones en *Android* es que los desarrolladores tienen acceso completo a los mismos *APIs* del *framework* usados por las aplicaciones base. Por esta razón mucho código es reutilizado, además, cualquier aplicación puede publicar sus capacidades.

2.2 MECANISMOS DE EXPORTACION DE DATOS DESDE UN DISPOSITIVO ANDROID

Por la naturaleza de los dispositivos móviles casi siempre es necesario reubicar la información recolectada, esto debido a la necesidad de contar con un mayor poder de computo, resguardar la data de la fragilidad del dispositivo , o simplemente compartir los datos con una aplicación externa.

Los mecanismos más conocidos para exportar datos desde un dispositivo con sistema operativo *Android* se describen a continuación. [8]

2.2.1 EXPORTACIÓN UTILIZANDO ARCHIVOS DE FORMATO CONOCIDO

La transmisión de los datos utilizando un archivo con un formato conocido e interpretado por un programa ya desarrollado. Este método consiste en generar una hoja de cálculo con una estructura bien conocida desde la aplicación *Android*, y luego manualmente cargar dicho archivo desde un computador. Las ventajas de este método son su fácil desarrollo, rápida implementación e independencia de redes de datos. Las desventajas son que el proceso de carga puede ser pesado para el dispositivo móvil y que el proceso está sujeto a errores ya que depende más del usuario.

2.2.2 EXPORTACIÓN UTILIZANDO HERRAMIENTAS NATIVAS DE ANDROID

La transmisión de los datos utilizando una sincronización vía USB (*Universal Serial Bus*) es un método que consiste en conectar el dispositivo a un computador, dicho dispositivo debe estar en modo de almacenamiento masivo o equivalente, de manera que la tarjeta de memoria sea visible, y, desde alguna aplicación externa se pueda acceder a los datos. La ventaja de este método es que, aunque hay cierto proceso manual, el contenido de los datos no es vulnerable como en el caso de la transmisión utilizando un archivo. La desventaja es que la necesidad de la conexión vía USB del dispositivo con el computador siempre que se quiera sincronizar.

2.2.3 EXPORTACIÓN UTILIZANDO INTERNET

Transmitir los datos utilizando un servidor en Internet es un método que consiste en enviar los datos desde la aplicación *Android* a un servidor con una dirección bien conocida. La ventaja de esta propuesta es que aumenta la accesibilidad de los datos y permite realizar respaldos frecuentes, mientras que la desventaja es que amerita que el dispositivo tenga conexión a internet, además, requiere tomar en cuenta aspectos de seguridad adicionales.

Por otro lado, el sistema operativo *Android* permite la conexión con diferentes servicios de almacenamiento en la nube como Drive, el cual funciona como un servicio de Google para cargar y descargar archivos de texto, audio, video, entre otros. Dicho servicio provee un API para el intercambio de datos, metadatos y archivos entre el servidor y el dispositivo [9].

Finalmente, luego de una investigación de los mecanismos de exportación de datos que están disponibles desde un dispositivo *Android*, se llegó a la conclusión que por la naturaleza del sistema a desarrollar y las necesidades de movilidad del usuario, la mejor opción para sincronizar los datos es utilizando Internet y un servidor de datos y almacenamiento bien conocido, el cual pueda ser accedido desde el dispositivo *Android* y desde la aplicación de escritorio.

2.3 MECANISMOS DE CAPTURA Y MANIPULACIÓN DE LA CAMARA EN ANDROID

Las cámaras digitales han evolucionado mucho en los últimos años, a tal punto, que es muy difícil encontrar algún dispositivo hoy en día que no posea una de estas. Por esta razón se han desarrollado mecanismos de captura y manipulación de las cámaras digitales en los dispositivos *Android*.

Tal como establece la documentación oficial de *Android* [2], el *Software Development Kit* de *Android* posee una biblioteca para el soporte de la comunicación con la cámara del dispositivo y permite ejecutar las funciones que la cámara brinda como capturar videos y fotos desde aplicaciones.

Las funciones principales del API son capturar fotos, capturar videos, presentar al usuario lo que está enfocando la cámara en ese momento, para realizar estas acciones en el dispositivo es necesario solicitar los siguientes permisos. Permiso de usar la cámara, permiso de usar las funcionalidades de la cámara, permiso de almacenamiento en el dispositivo, permiso de grabar audio y opcionalmente el permiso de saber la localización del dispositivo.

Para utilizar el API de *Android* que maneja la cámara, es posible realizarlo por dos vías, la primera, es utilizar una aplicación existente que controle la cámara, o, por otro lado, implementar una aplicación que manipule directamente la cámara del dispositivo.

Cuando se delega a una aplicación existente que controle la cámara del dispositivo, se realiza una solicitud de capturar una foto o video por medio de la aplicación que por defecto tiene el sistema para el control de la cámara del dispositivo, esta aplicación toma control del teléfono y luego de que se complete ese proceso retorna el control a la aplicación desarrollada.

Por otro lado, más relacionado a lo necesario para el proyecto, la creación de una aplicación que controle la cámara del dispositivo directamente requiere un desarrollo más profundo pero provee más libertad para el desarrollador de la aplicación para acoplar las funcionalidades de la cámara con la interfaz de la aplicación, logrando así una experiencia de usuario más confortable.

Algunos pasos necesarios para crear una aplicación que controle directamente la cámara del dispositivo son, detectar y acceder a la cámara, en el código de la aplicación debe existir un segmento que detecte la cámara y luego solicite acceso, crear una clase *Preview*, crear una clase que maneje la vista preliminar de la cámara basándose en implementaciones de las bibliotecas del sistema operativo, construir un *Preview Layout*, utilizado para mostrar en la interfaz lo que genere la clase de vista preliminar de la cámara, desarrollar interfaz de captura, crear botones y demás características que permitan al usuario tomar decisiones sobre la cámara, capturar y guardar archivos, codificar las funcionalidades que le permiten a la aplicación capturar y guardar en el dispositivo las imágenes y videos producto de la captura de la cámara, y liberar la cámara, luego de usar la cámara es necesario que la aplicación la libere correctamente para que pueda ser utilizada por otras aplicaciones [10].

2.4 MECANISMOS DE TEMPORIZACIÓN EN DISPOSITIVOS ANDROID

En muchas disciplinas existe la necesidad de cronometrar eventos, por lo que a lo largo de la historia se han creado dispositivos que se dedican a esta labor. *Android* por su parte posee un API que permite hacer esta labor muy fácilmente, permitiendo así el desarrollo de aplicaciones de alto nivel que pueden ser utilizadas por cualquier persona con un dispositivo que tenga este sistema operativo.

El API de *Android*, permite utilizar funciones para ejecutar segmentos de código según un temporizador, ciertos intervalos y rangos de tiempo. Esta característica la brinda la clase *CountDownTimer* descrita en la documentación oficial de *Android* [11], la cual forma parte del conjunto de clases que contiene el sistema operativo.

2.5 CONEXIÓN A BASE DE DATOS DESDE ANDROID

Una gran parte de las aplicaciones tiene como insumo principal una fuente de datos externa a ella, por lo que se necesita realizar una conexión a una base de datos, *Android* no escapa de esta realidad, ya que debido a las limitaciones físicas es muy difícil tener un gran volumen de datos en un dispositivo móvil con características comunes, por esto, realizar conexiones a servidores de bases de datos externos a un dispositivo es una acción muy común cuando se desarrolla bajo esta tecnología.

Las aplicaciones en *Android* se pueden realizar utilizando Java, adicionalmente este lenguaje permite una sencilla comunicación con la base de datos utilizando un objeto de la clase `java.sql`, utilizando una base de datos SQLite, esta conexión se puede realizar de la forma mostrada en la Ilustración 1.

```
import java.sql.*;

public class SQLiteJDBC
{
    public static void main( String args[] )
    {
        Connection c = null;
        try {
            Class.forName("org.sqlite.JDBC");
            c = DriverManager.getConnection("jdbc:sqlite:test.db");
        } catch ( Exception e ) {
            System.err.println( e.getClass().getName() + ": " + e.getMessage() );
            System.exit(0);
        }
        System.out.println("Opened database successfully");
    }
}
```

Ilustración 1 CONEXIÓN A BASE DE DATOS SQLITE DESDE JAVA

2.6 HERRAMIENTAS DE DESARROLLO

Existen una gran cantidad de herramientas para desarrollar aplicaciones, sin embargo, es de particular interés las que están relacionadas con el lenguaje de programación Java, ya que permite desarrollar aplicaciones multiplataforma de escritorio y móviles además de tener una documentación extensa.

2.6.1 LENGUAJE DE PROGRAMACIÓN JAVA

Según la documentación oficial, lo definen como un lenguaje de programación multipropósito, concurrente, basado en una estructura de clases y orientado a objetos. Está diseñado para ser lo suficientemente simple para ser utilizado por la mayor cantidad de programadores posible. Posee relaciones con lenguajes como C y C++, con organización

diferente y aspectos heredados de otros lenguajes de programación. Pensado como un lenguaje de producción, más que un lenguaje de desarrollo [12].

El lenguaje de programación Java permite un alto grado de portabilidad ya que 97% de los computadores de oficina permiten la ejecución de programas desarrollados utilizando este lenguaje aunque dichos computadores utilicen sistemas operativos diferentes. Además, posee una gran comunidad desarrolladores, suficiente documentación del lenguaje y sus bibliotecas, así como la posibilidad de escoger poderosos entornos de desarrollo para diseñar y desarrollar aplicaciones [13].

Este lenguaje permite el desarrollo de aplicaciones tanto móviles, como de escritorio, así como un vasto conjunto de aplicaciones más específicas. Es un lenguaje ideal para adaptarse a cambios de plataforma y dispositivos, además de que contiene un conjunto de bibliotecas y paquetes de desarrollo que permiten desde cálculo matemático hasta conexiones a base de datos utilizando las facilidades que brinda el sistema operativo del dispositivo.

Existe un conjunto de IDEs que permiten desarrollar utilizando el lenguaje Java, entre estos está NetBeans y Eclipse como los principales por su comunidad, bibliotecas que apoyan el desarrollo, organización óptima de la interfaz y licencia de libre uso.

2.6.1.1 CONEXIÓN A BASE DE DATOS MYSQL DESDE JAVA

Según la documentación oficial de Java [13], la conexión de base de datos a MySQL se realiza utilizando un JDBC. Este conector JDBC lo provee MySQL para diferentes sistemas operativos, normalmente puede ser descargado de la página oficial de MySQL. El JDBC permite la conexión a los orígenes de datos mediante una URL y los parámetros de autenticación. En la Ilustración 2, se presenta un ejemplo en código Java de la utilización del conector a MySQL.

```

public Connection getConnection() throws SQLException {

    Connection conn = null;
    Properties connectionProps = new Properties();
    connectionProps.put("user", this.userName);
    connectionProps.put("password", this.password);

    if (this.dbms.equals("mysql")) {
        conn = DriverManager.getConnection(
            "jdbc:" + this.dbms + "://" +
            this.serverName +
            ":" + this.portNumber + "/",
            connectionProps);
    } else if (this.dbms.equals("derby")) {
        conn = DriverManager.getConnection(
            "jdbc:" + this.dbms + ":" +
            this.dbName +
            ";create=true",
            connectionProps);
    }
    System.out.println("Connected to database");
    return conn;
}

```

Ilustración 2 CONEXIÓN A BASE DE DATOS MYSQL DESDE JAVA

2.6.2 LENGUAJE DE PROGRAMACIÓN PHP

La documentación oficial de PHP [14], lo define como un lenguaje de programación de código abierto y ampliamente popular entre los desarrolladores web, el cual puede ser incrustado dentro del código HTML y utiliza la teoría cliente-servidor.

El campo principal de PHP es ser ejecutado como lotes de *scripts* del lado de servidor, con este objetivo hay 3 actores que entran en funcionamiento para un proceso exitoso, el analizador de PHP o también llamado módulo CGI del servidor, un servidor web y un navegador web. Es necesario que el servidor web tenga una instalación de PHP conectada, para acceder al resultado del *script* es necesario ejecutarlo por medio del servidor web desde un navegador.

Una característica favorable del lenguaje es que puede ser ejecutado en Linux, muchas variantes de Unix, Microsoft Windows, Mac OS X, RISC OS, entre otros sistemas operativos. Adicionalmente PHP es compatible con la mayoría de los servidores web de hoy en día, incluyendo Apache, IIS y muchos otros.

En cuanto a los aspectos de programación, este lenguaje permite programar utilizando el paradigma de programación por procedimientos o programación orientada a objetos, inclusive una mezcla de ambos. No solo permite escribir código HTML, también imágenes, ficheros PDF, archivos Flash, XHTML, JSON o cualquier archivo XML.

En relación a la conexión de base de datos, PHP soporta un amplio abanico de base de datos. Conectarse a cualquier base de datos que admita el estándar de conexión abierta a base de datos por medio de la extensión ODBC es muy sencillo utilizando este lenguaje. Adicionalmente es muy poderoso para conectarse a servicios locales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (Windows), entre otros. El manejo de *sockets* se utiliza tanto puros para conexiones de red con protocolos conocidos, como para comunicarse utilizando cualquier otro protocolo.

2.6.2.1 CONEXIÓN A BASE DE DATOS MYSQL DESDE PHP

Para conectar a una base de datos MySQL, PHP brinda una extensión denominada *mysqli*, la cual permite acceder a la funcionalidad proporcionada por MySQL 4.1 y posteriores, esta viene incluida en las versiones de PHP 5 y posteriores, los principales beneficios que brinda esta extensión son, una interfaz orientada a objetos, soporte para declaraciones preparadas, soporte para múltiples declaraciones transacciones, opciones de depuración mejoradas y soporte para servidor empujado. En la Ilustración 3 se muestra un segmento de código como ejemplo de una conexión a MySQL utilizando el lenguaje de programación PHP.

```
<?php
$mysqli = new mysqli("ejemplo.com", "usuario", "contraseña",
"basedatos");
if ($mysqli->connect_errno) {
    echo "Fallo al conectar a MySQL: " . $mysqli->connect_error;
}
$resultado = $mysqli->query("SELECT 'elecciones para complacer
a todos.' AS _msg FROM DUAL");
$fila = $resultado->fetch_assoc();
echo $fila['_msg'];
?>
```

Ilustración 3 CONEXIÓN A BASE DE DATOS MYSQL DESDE PHP

2.6.3 HERRAMIENTAS DE DESARROLLO EN ANDROID

Las principales herramientas de desarrollo recomendadas en la página oficial de *Android* son NetBeans [15] y Eclipse [16], ambas de acceso libre. NetBeans puede usar un

complemento llamado *NBAndroid* para mejorar sus capacidades en el desarrollo para esa plataforma y Eclipse tiene disponible ADT (*Android Development Tools*) para el mismo objetivo.

2.6.3.1 IDE ECLIPSE

El entorno de desarrollo integrado Eclipse es una herramienta informática de código abierto multiplataforma diseñada para construir Entornos de Desarrollo Integrado, el cual fue originalmente desarrollado por Jim Des Riveres (*IBM*) [16].

Las características especiales de esta herramienta son poseer un *kit* de Desarrollo de *Software*, la capacidad de agregar complementos al ambiente de desarrollo y de poseer herramientas que facilitan el desarrollo en Java.

La plataforma de Eclipse está conformada por:

- Plataforma principal, incluye búsqueda, integración y ejecución de complementos de desarrollo y demás características principales de Eclipse.
- OSGi (*Open Services Gateway Initiative*), utilizado para la construcción estándar de *Software*.
- SWT (*Standard Widget Toolkit*), Biblioteca de componentes de interfaz gráfica en Java desarrollado en Eclipse.
- JFace, Conjunto de componentes de interfaz gráfica que fueron desarrollados por IBM para Eclipse.
- *Workbench* de Eclipse, interfaz principal de trabajo compuesta por editor de texto, vistas, perspectivas, asistentes de desarrollo, entre otros.

ADT (*Android Development Tools*) es un componente para el entorno de desarrollo integrado diseñado para brindar un poderoso ambiente integrado en el cual se pueden construir aplicaciones *Android*. Actualmente en la versión 23.0.3 Agosto de 2014 según [7].

Este componente de desarrollo de *Software* agrega capacidades para construir proyectos *Android* rápidamente, diseñando su interfaz de usuario y agregando paquetes del API de *Android*.

2.6.3.2 IDE NETBEANS

Según la documentación oficial de NetBeans [15], es uno de los IDE más ampliamente utilizados a nivel mundial para desarrollo en Java por su versatilidad e intuitivo manejo de los distintos tipos de archivos y herramientas que pueden ser necesitados durante el ciclo de desarrollo de una aplicación.

Uno de los motivos por el que NetBeans es ampliamente utilizado es la increíble facilidad que existe para crear interfaces gráficas de usuario usando esta herramienta, esto se debe a que NetBeans permite visualizar en tiempo de diseño la interfaz, así como también *JavaFx*, que posee un conjunto muy rico de paquetes gráficos y multimedia que desde otro IDE tendrían que ser obtenidos mediante una instalación externa.

NetBeans es un IDE diseñado principalmente para el desarrollo en el lenguaje de programación Java, aunque también es ampliamente utilizado para el desarrollo web adicionándole ciertos componentes extras y además es de código abierto. Actualmente se encuentra en su versión 8.0.

Lo que diferencia a NetBeans del resto de los IDE es que está a muy alto nivel, es decir, brinda herramientas intuitivas para casi cualquier cosa que se desee hacer dentro de él, cosa, que en algún otro puede ser engorroso.

NBAndroid es un *plugin* diseñado para NetBeans que permite la creación y desarrollo de proyectos en tecnología *Android*.

2.6.3.3 IDE ANDROID STUDIO

Según la documentación oficial de *Android Studio* [17], es el IDE oficial para el desarrollo de aplicaciones que se ejecuten en el sistema operativo *Android*, permite compilar, construir, depurar, escribir y ejecutar aplicaciones en emuladores de sistema operativo *Android*.

Cuenta con una integración de herramientas como, *Gradle-based*, la cual mejora la compilación de las aplicaciones, adicionalmente, el IDE *Android Studio* agrega soporte en una documentación cuantiosa que permite al desarrollador apoyarse para completar los requerimientos de las aplicaciones comunes.

Facilita la edición de interfaces de la aplicación, además de agregar un editor de texto enriquecido de alta eficiencia y estabilidad. Además, agrega un componente denominado *ProGuard* que reduce, optimiza y ofusca el código escrito para el desarrollo de la aplicación.

No necesita instalar ningún *plugin* para poder desarrollar en *Android*, viene completamente preparado para el desarrollo y hasta opcionalmente permite descargar e instalar la biblioteca SDK junto con el IDE.

2.6.3.4 SDK ANDROID

Sin importar el IDE también se debe trabajar con el SDK (*Software Development Kit*) de *Android* que no son más que un conjunto de herramientas que permiten el desarrollo sobre esta tecnología.

La documentación oficial de *Android* SDK [7] lo define como un conjunto de herramientas para el soporte del desarrollo de aplicaciones móviles para el sistema operativo *Android*, que son clasificadas en dos grupos *SDK Tools* y *Platform Tools*. La diferencia entre estos dos grupos es la frecuencia de actualización para no impactar negativamente el desarrollo. El *SDK Tools* está conformado por un emulador *Android*, un servidor para procesos *Dalvik* utilizado para procesos de depuración de código, un componente de *SQLite3*, entre otros.

2.6.4 API DE GOOGLE DRIVE

Según la documentación oficial del API de Google Drive [9], lo definen como una biblioteca que permite el acceso a los servicios de contenidos que ofrece Drive con su respectiva cuenta de Google, permite la administración de archivos y su *metadata*.

Para utilizar los servicios que brinda el API de Google Drive es necesario utilizar la biblioteca de Google Play Services SDKs y generar un certificado de acceso a la aplicación desde la cuenta Drive que la utilizará.

La autenticación con los servicios de Google es necesaria para utilizar cualquier funcionalidad que brinda el API de Google Drive, luego de esta autenticación cruzada es posible solicitar los servicios de creación de archivos, edición del contenido, edición de la *metadata* de los archivos, descarga de archivos, administración de carpetas, entre otros.

Esta biblioteca está disponible para *Android*, pero también tiene implementaciones que pueden ser utilizadas en servicios web, aplicaciones web, aplicaciones de escritorio, entre otros. Adicionalmente, está disponible para varios lenguajes de programación, incluyendo Java.

2.6.5 JSON

El estándar ECMA-404 [18], define JSON como un formato de texto que facilita el intercambio de datos estructurados entre todos los lenguajes de programación. La sintaxis de JSON está compuesta de aparatos ortopédicos, soportes, dos puntos y comas que son útiles en muchos contextos, perfiles y aplicaciones.

JSON depende de *Unicode* por los códigos utilizados en la notación, ya que el texto que forma una cadena JSON es una secuencia de *tokens* formados por códigos *Unicode* que conforman la gramática de valores JSON.

En la Ilustración 4 se puede observar la estructura básica de un objeto JSON según como lo describe el estándar ECMA-404.

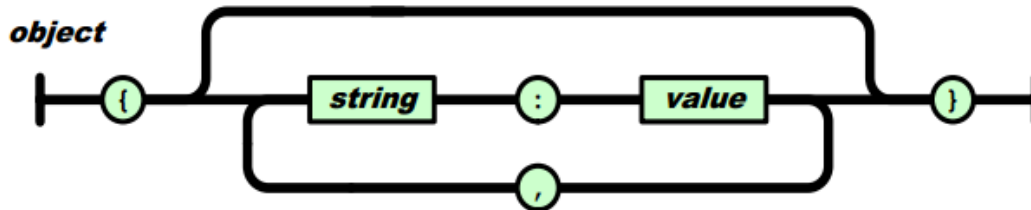


Ilustración 4 ESTRUCTURA DE UN OBJETO JSON

2.6.6 GSON

Los documentos publicados por Google [19] definen a GSON como una biblioteca codificada en Java que puede ser usada para convertir Objetos Java en su representación en JSON, de igual forma puede convertir cadenas JSON a su equivalente Objeto Java. Entre sus características principales están:

- Provee métodos simples para la codificación y decodificación de cadenas JSON.
- Amplio soporte a estructuras genéricas de Java.
- Permite representaciones complejas de objetos.
- Amplia documentación del API y guías de usuario.

La biblioteca para manejo de objetos JSON de *Google*, GSON es muy utilizada y documentada, en la Ilustración 5 podemos ver un ejemplo en lenguaje Java del uso de la biblioteca y algunos de sus métodos.

```
Gson gson = new Gson(); // Or use new GsonBuilder().create();
MyType target = new MyType();
String json = gson.toJson(target); // serializes target to json
MyType target2 = gson.fromJson(json, MyType.class); // deserializes json into target2
```

Ilustración 5 EJEMPLO DE USO DE LA BIBLIOTECA GSON

2.6.7 BIBLIOTECA CHARTS4J

Tal como está detallado en su documentación [20], charts4j es una biblioteca para la generación de gráficos codificada en el lenguaje de programación Java y diseñada para ser utilizada bajo este mismo lenguaje, utiliza el *API* de *Google* para el cálculo y construcción del gráfico utilizando peticiones bajo el protocolo HTTP.

Esta biblioteca permite solicitar, generar y visualizar casi todos los gráficos que la biblioteca de Google permite, varias decenas, los cuales utilizan tecnología HTML5/SVG. Como consecuencia de esto, aumenta la velocidad de despliegue y disminuye significativamente los recursos de procesamiento necesarios en el computador para generar los gráficos de alta calidad visual, sin embargo, ya que está basado en el paradigma cliente-servidor, el protocolo HTTP y *Cloud Computing* es necesario una conexión a Internet para poder utilizarla, sin conexión a Internet es imposible solicitar, generar o visualizar los gráficos; otra desventaja de la biblioteca es que las solicitudes HTTP viajan por GET, es decir, en la URL por lo tanto tiene un límite de datos que pueden ser graficados, para algunos sistemas esto no es problema ya que el límite es bastante alto, unos 2000 registros de datos, pero es posible codificar los datos de tal manera de disminuir la cantidad de registros sin perder la visualización de los datos, solo disminuye un poco el nivel de detalle al ser visualizados.

La licencia de publicación y uso de la biblioteca es *MIT License*, la cual permite su utilización, copia, modificación, fusión, publicación, distribución, sublicenciar y venta de forma gratuita.

Debido a que está programada y diseñada para el lenguaje de programación Java, la biblioteca utiliza el paradigma de programación orientada a objetos, como se muestra en la Ilustración 6, adicionalmente cuenta con una detallada documentación de su instalación, funcionamiento, clases y métodos [21].

```
// EXAMPLE CODE START
// Defining lines
final int NUM_POINTS = 25;
final double[] competition = new double[NUM_POINTS];
final double[] mywebsite = new double[NUM_POINTS];
for (int i = 0; i < NUM_POINTS; i++) {
    competition[i] = 100-(Math.cos(30*i*Math.PI/180)*10 + 50)*i/20;
    mywebsite[i] = (Math.cos(30*i*Math.PI/180)*10 + 50)*i/20;
}
Line line1 = Plots.newLine(Data.newData(mywebsite), Color.newColor("CA3D05"), "My Website.com");
line1.setLineStyle(LineStyle.newLineStyle(3, 1, 0));
line1.addShapeMarkers(Shape.DIAMOND, Color.newColor("CA3D05"), 12);
line1.addShapeMarkers(Shape.DIAMOND, Color.WHITE, 8);
Line line2 = Plots.newLine(Data.newData(competition), SKYBLUE, "Competition.com");
line2.setLineStyle(LineStyle.newLineStyle(3, 1, 0));
line2.addShapeMarkers(Shape.DIAMOND, SKYBLUE, 12);
line2.addShapeMarkers(Shape.DIAMOND, Color.WHITE, 8);
```

Ilustración 6 SUBCONJUNTO DE CÓDIGO DE GRAFICO UTILIZANDO CHARTS4J API

2.6.8 SUGAR ORM

Publicado y descrito como un proyecto en el sitio web *GitHub* [22], consiste en una biblioteca que provee una interfaz de comunicación limpia y simple entre una aplicación *Android* y una base de datos *SQLite*, lo cual permite minimizar la escritura de código SQL para la comunicación con la base de datos.

La biblioteca da soporte a la creación de la base de datos, además, se encarga de las relaciones entre las tablas y provee una API simple y limpia para las operaciones. Todo esto lo realiza utilizando un modelo basado en el paradigma de programación orientada a objetos, como se puede visualizar en la Ilustración 7.

```
Book book = Book.findById(Book.class, 1);  
book.title = "updated title here"; // modify the values  
book.edition = "3rd edition";  
book.save(); // updates the previous entry with new values.
```

Ilustración 7 BÚSQUEDA Y MODIFICACIÓN DE UN OBJETO DEL MODELO DE DATOS UTILIZANDO SUGAR ORM

Fue desarrollado pensando en aquellas aplicaciones de plataforma *Android* que necesitan soportarse bajo un modelo de datos grande y complejo, por lo tanto el objetivo principal de la biblioteca es almacenar, obtener e iterar sobre los objetos del negocio disminuyendo la complejidad de manipular grandes repositorios de datos. Por otro lado, una característica que beneficia mucho al desarrollador es que la biblioteca está diseñada para disminuir el tiempo en aspectos técnicos y aumentar el tiempo productivo desarrollando la lógica del negocio.

2.6.9 SERVIDOR HTTP APACHE

Los documentos oficiales de la fundación Apache [23], lo definen como un poderoso y flexible servidor HTTP, el cual implementa los últimos protocolos de comunicación web, incluyendo HTTP.

Una ventaja a tomar en cuenta es su alta extensibilidad aplicando módulos de terceros, dichos módulos pueden ser escritos utilizando un *API* que provee, y el hecho de ser altamente configurable por el usuario.

Posee licencia de código abierto y uso no restringido, por otro lado, puede ser ejecutado en la mayoría de los sistemas operativos y es actualizado constantemente. Entre las características más importantes están, almacenamiento de registros de uso y errores, distinguir entre un dispositivo virtual y uno real, negociación del contenido, múltiples directivas de indexación de carpetas, proteger archivos con claves, entre otros.

2.6.10 SHA-256

El RFC 6234 [24] describe algunos de los algoritmos publicado por el NIST en el año 2008, en específico el algoritmo de *SHA-256* permite generar una cadena de 256 bits como resultado de un proceso matemático sobre un mensaje con una longitud variable. Esta función es direccional y resistente a las colisiones, lo que significa que solo puede ser

calculada desde la fuente (el mensaje), a la cadena de bits resultante; además, dicho resultado no será repetido por distintas fuentes.

2.6.11 MP4PARSER

M4PARSER es un proyecto patrocinado por Google, que funciona como una biblioteca escrita en java que sirve para manipular archivos .mp4 , además , a través de ella se pueden crear videos compuestos de otros, es decir, fácilmente se puede mezclar dos videos mp4 en uno solo [25].

La ventaja principal de mp4parser es lo fácil que es de usar además de la amplia cantidad de documentación disponible, por otro lado su principal defecto es la imposibilidad para codificar archivos desde cero, es decir, es imposible crear un archivo mp4 sin tener alguna fuente.

2.6.12 HERRAMIENTAS SELECCIONADAS

Siguiente a la investigación realizada sobre las herramientas disponibles para soportar el desarrollo de las aplicaciones y la arquitectura que soportará el funcionamiento del sistema, se seleccionó un conjunto de herramientas y tecnologías que permitieran un rápido, eficiente y efectivo desarrollo.

En relación a la aplicación para el sistema operativo *Android*, fue desarrollado utilizando el IDE *Android Studio*, al cual se le agrego la biblioteca SDK que ofrece *Android*, adicionalmente se seleccionó *Sugar ORM* como biblioteca para facilitar el trabajo con la base de datos de *SQLite*, donde se almacenará los datos que manejará la aplicación del dispositivo móvil. En la interfaz que captura el combate fue necesario utilizar métodos del manejo de la cámara del sistema operativo, para controlar la interfaz en su totalidad mientras se capturan los eventos. Luego de capturar los combates, relacionado a la sincronización de los datos se escogió el método de sincronización utilizando un servidor conocido en Internet, para los datos fue seleccionado el protocolo HTTP para realizar peticiones a un servidor Web bien conocido utilizando mensajes codificados con estructura JSON, para esto fue eligió la biblioteca de Google GSON, por otro lado, para la sincronización del material multimedia del combate se optó por utilizar la biblioteca de *Google Drive*, la cual permite realizar conexiones y subir archivos desde el dispositivo *Android*.

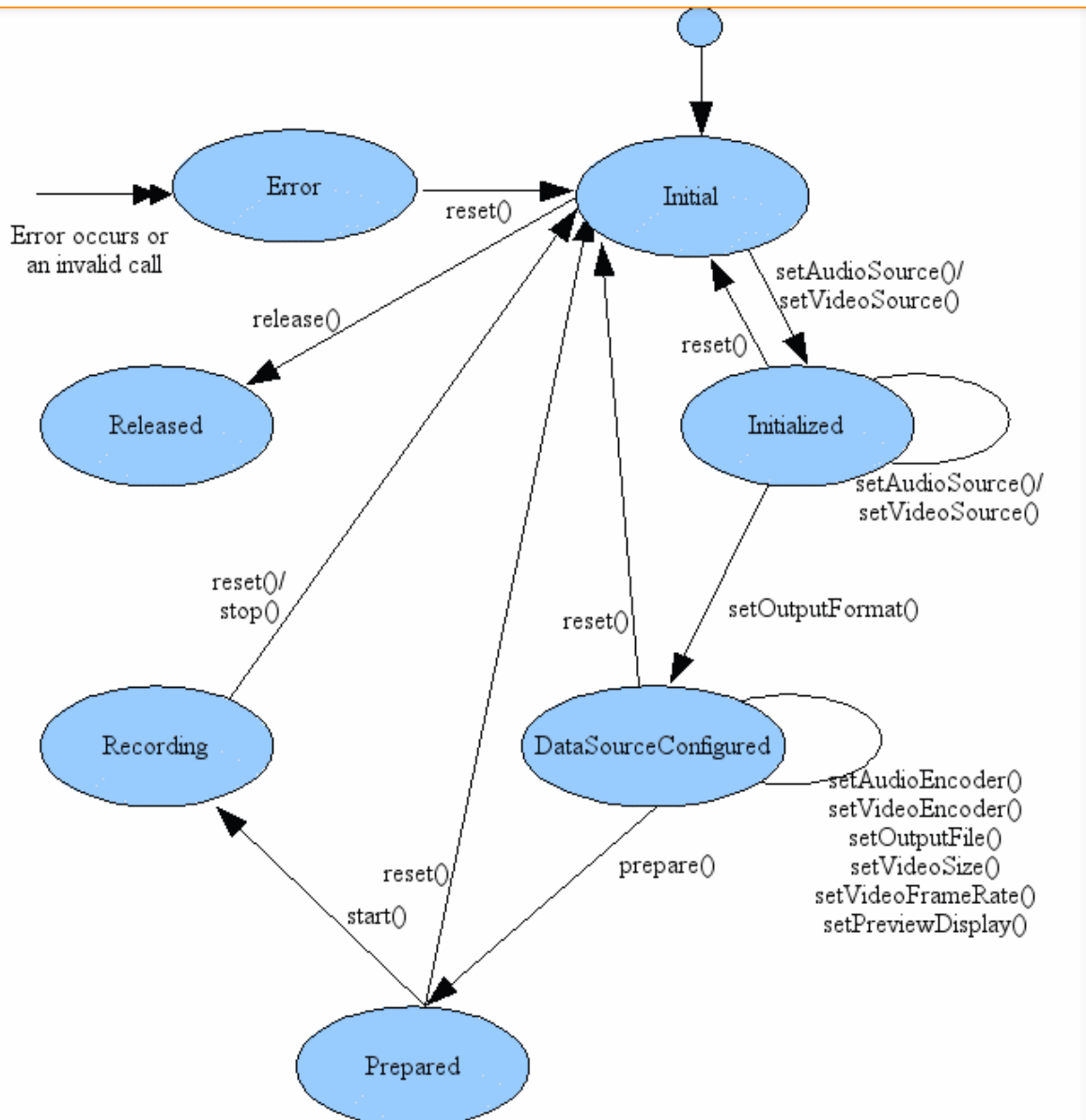


Ilustración 8 MAQUINA DE ESTADO DE LA CLASE MEDIARECORDER [26]

En cuanto a los videos, como se puede ver en la Ilustración 8, la clase *MediaRecorder* de *Android* no posee la capacidad de pausar la grabación por lo que era necesario idear un mecanismo que permitiera mezclar todos los videos que podría generar un combate cada vez que se usara la funcionalidad de pausar, para esto, se utilizó la biblioteca *mp4parser* y un esquema de "Buffer" mediante el cual cada vez que deba ser generado un segundo archivo de video este será *buffer.mp4*, si este ya existe entonces será mezclado con el primer archivo creado y luego será sobrescrito.

Para responder las peticiones HTTP, funcionando como una interfaz de comunicación con el modelo de datos, se seleccionó un servidor HTTP *Apache* con capacidad de procesar lenguaje PHP, los segmentos de código en dicho lenguaje tendrían

el objetivo de recibir, traducir, procesar y almacenar la información enviada desde el dispositivo *Android* a la base de datos MySQL en un servidor bien conocido. Por otro lado, en caso de que el dispositivo *Android* solicite actualizar una entidad, la interfaz tiene el objetivo de enviar la información que haya sido modificada o agregada de la base de datos al dispositivo *Android*. Adicionalmente, se utilizó *SHA-256* para asistir en características de autenticación con clave PSK.

En el caso de la aplicación de escritorio, se seleccionó la biblioteca de *Google Drive* para sincronizar los videos y una conexión directa con el servidor *MySQL* para almacenar y cargar los datos de los combates y combatientes, en relación a la generación de gráficos se optó por la biblioteca *charts4j* que es un proyecto apoyado y soportado por *Google*.

2.7 MECANISMOS DE ALMACENAMIENTO

En la actualidad es común la recopilación de datos de diversas fuentes, para poder almacenar los datos de forma segura, organizada y accesible se utilizan diversos mecanismos de almacenamiento como lo son las tecnologías *SQLite*, *MySQL* y *Google Drive*. Las cuales fueron seleccionadas para el desarrollo de este proyecto y se describen brevemente a continuación.

2.7.1 SQLITE

La organización de *SQLite* lo define como una biblioteca que implementa un motor de base de datos SQL transaccional auto contenido, sin servidor y sin configuración inicial. Su código fuente de dominio público y es libre de uso para cualquier propósito, tanto privado como comercial. La biblioteca lee y escribe directamente al disco, no utiliza un proceso servidor aparte. Contiene las herramientas de una base de datos SQL, como son tablas, índices, disparadores, vistas y relaciones [27].

2.7.2 MYSQL

La documentación oficial de *MySQL* [28] lo define como una solución de *Software* para servidor de base de datos SQL multi-usuario y multi-hilo. Posee una licencia abierta GNU de público general, o una licencia comercial que puede ser comprada.

MySQL es un sistema manejador de base de datos relacional de código abierto de respuesta rápida, seguro, escalable y de fácil uso. Adicionalmente, trabaja bajo el modelo cliente-servidor, por lo tanto es necesario instalar, configurar y ejecutar un servidor para su correcto funcionamiento.

Los tipos de datos que maneja son:

- FLOAT
- DOUBLE
- CHAR
- VARCHAR
- BINARY
- VARBINARY
- TEXT
- BLOB
- DATE
- TIME
- DATETIME
- TIMESTAMP
- YEAR
- SET
- ENUM

Posee un conjunto de bibliotecas para la conexión desde diferentes lenguajes de programación, como son: C, C++, Java, Perl, PHP, Python, Ruby, C#, entre otros.

2.7.3 GOOGLE DRIVE

Basado en los documentos publicados por *Google* [29], se puede definir como un servicio de almacenamiento de archivos en la nube ofrecido por *Google* desde el año 2002 para los usuarios de sus servicios más utilizados. Actualmente ofrece 15 GB de espacio de almacenamiento gratuito que es compartido entre los servicios de *Google Drive*, *Gmail* y las fotos de *Google+*.

Para acceder al servicio solo es necesario una cuenta de correo *Gmail*, luego del registro el servicio permite buscar, compartir, crear, añadir o subir archivos a los servidores de *Google Drive*. Adicionalmente, la plataforma de *Drive* permite acceder al archivo desde casi cualquier localización que disponga de una conexión a Internet. Por otro lado, la autenticación que brinda el servicio de *Gmail* permite asegurar el acceso a los archivos almacenados.

Existen diferentes aplicaciones, para teléfonos y computadores de escritorio que permiten acceder a los archivos almacenados en los servidores de *Drive*. En general, dichas aplicaciones requieren poder de cálculo y procesamiento bastante austeros, ya que parte de la carga de cómputo es realizado por el servidor.

2.8 EVENTOS Y REGLAS DE LOS COMBATES DE JUDO

El Judo según la Federación Internacional de Judo es el segundo deporte más practicado del mundo, está basado en lanzar a los oponentes o derrumbarlos, usar los puntos de presión o estrangulaciones momentáneas, también en algunos tipos de prácticas de esta disciplina está permitido el uso de ciertos golpes con las manos [1].

Es un deporte olímpico y según investigaciones recientes aporta al judoca una mejora en aspectos psicológicos y físicos como la autoestima, capacidad de auto control, psicomotricidad además de fortalecer los huesos y activar el metabolismo , por esto es el deporte olímpico más recomendado por los médicos para que sea practicado por los niños [1].

Según indica la normativa de la Federación Internacional de Judo [30], los combates de Judo son de 4 a 5 minutos, se realizan en un ambiente controlado, con un uniforme específico, siguiendo reglas y bajo la estricta vigilancia de al menos 3 árbitros.

Adicionalmente, la Federación Internacional de Judo también deja claro en las normativas el área de combate, la cual está formada por dos zonas, una interna de entre 8 metros por 8 metros y máximo 10 metros por 10 metros, adicionalmente una franja de seguridad de al menos 3 metros de ancho, tal como se indica en la Ilustración 9.



Ilustración 9 ÁREA DE COMPETENCIA DEL JUDO [30]

2.8.1 REGLAS DEL JUDO

A continuación las reglas vigentes en los combates de Judo entre 2014-2016 descritas por la FIJ (Federación Internacional de Judo) [30]:

- El área de competencia deberá ser de 14 metros por 14 metros, de un material especial denominado *Tatamis*, Dividido en dos zonas, una interna más pequeña que se denomina zona de combate es de unos 8 metros por 8 metros mínimo y máximo 10 metros por 10 metros, la zona externa denominada zona de seguridad deberá ser de al menos 3 metros de ancho, tal como se puede observar en la figura 6.
- Por cada área de competencia deberán estar dos tablas de puntaje que mostrarán los puntos de forma horizontal y externo al área de competencia. La tabla de puntos debe mostrar las penalizaciones recibidas por los competidores.
- El competidor deberá portar uniforme de Judo cumpliendo las condiciones que emanan las reglas.
- El competidor deberá respetar las reglas de higiene establecidas por el jurado.
- En los eventos internacionales la competencia es conducida por tres árbitros, los cuales son de distinta nacionalidad que los competidores, los tres deben cumplir con el reglamento de vestimenta y se comunican utilizando radios, adicionalmente son asistidos con el puntaje, tiempo y demás notas necesarias por personal de apoyo.
- El árbitro principal estará dentro del área de combate, el deberá conducir el combate y administrar las decisiones, adicionalmente deberá ser garante que las decisiones del combate sean recordadas.
- Dos árbitros estarán sentados en la mesa conectados por radio con el árbitro principal y podrán ser asistidos con videograbaciones del combate, el sistema de acuerdo será la mayoría de 3.
- Al momento de ocurrir algún evento el árbitro deberá hacer algún gesto que indique que ha ocurrido un evento.
- La competencia deberá realizarse dentro del área de combate. El lanzamiento deberá ser iniciado cuando ambos competidores estén en el área de combate, toda técnica realizada fuera del área de combate no será reconocida. Todas las acciones serán válidas y podrán continuar siempre y cuando parte del cuerpo del combatiente toque el área de combate.
- La duración del combate deberá regirse por el reglamento del torneo.
- El tiempo en el cual el árbitro anuncia los eventos ocurridos no deberá ser contado en el tiempo de la competencia.
- El final del tiempo asignado deberá ser indicado por el árbitro con una campana u otra señal audible similar. Cuando se utilizan varias áreas de competencia al mismo tiempo el sonido de final de tiempo asignado deberá ser diferente.
- Si al sonar la señal de final del tiempo asignado el competidor está realizando alguna técnica de *Osaekomi*, se extenderá el tiempo de la competencia hasta que termine de realizar la técnica.
- Cualquier resultado de una técnica iniciada simultáneamente con la señal de final de tiempo es válida. Cualquier resultado de una técnica iniciada posterior a la señal de final de tiempo es inválida.
- Los árbitros, tanto principal como los de mesa, deberán estar preparados en posición antes de la entrada de los competidores al área de combate.

- Los combatientes podrán pasar de combate en pie a combate en el suelo.
- El árbitro podrá detener el combate temporalmente anunciando un *Mate*, los combatientes deberán regresar rápidamente a la posición inicial.
- El cualquier caso en el cual el árbitro detenga temporalmente el combate sin causar cambio en las posiciones, deberá anunciar un *Sono-mama*, esto solo podrá ser aplicado en situaciones que el combate se realice en el suelo.
- El árbitro deberá anunciar el fin del combate anunciando *Sore-made*.
- El árbitro deberá anunciar *Ippon* cuando en su opinión la técnica corresponda, siguiendo el reglamento del torneo.
- Cuando un combatiente recibe un segundo *Waza-ari* en el combate el árbitro deberá anunciar *Waza-ari-awasete-Ippon*.
- El árbitro deberá anunciar *Waza-ari* cuando en su opinión la técnica haya sido realizada por uno de los competidores, siguiendo el reglamento del torneo.
- El árbitro deberá anunciar *Yuko* cuando en su opinión la técnica haya sido realizada por uno de los competidores, siguiendo el reglamento del torneo.
- El árbitro deberá anunciar *Osaekomi* cuando en su opinión la técnica haya sido realizada por uno de los competidores, siguiendo el reglamento del torneo.
- Los actos prohibidos están divididos en Ofensas y Severas. Estas tendrán las penalizaciones que estén en el reglamento.
- Si un combatiente no aparece para el combate luego de tres llamados en 1 minuto de intervalo perderá el combate.
- La decisión si un combatiente no puede continuar por lesión, dolencia o accidente durante el combate la tendrá el árbitro siguiendo las reglas del torneo.

2.8.2 EVENTOS EN LOS COMBATES DE JUDO

Los eventos que destacan en el Judo y reflejan una mayor cantidad de puntos en la competencia, según la Federación Internacional de Judo [30], son:

- *Ippon*, ocurre mayormente cuando el oponente es proyectado y cae sobre su espalda concluyendo la pelea y dando como ganador al proyectante, tal como se muestra en la Ilustración 10, por otro lado, también califica Ippon inmovilizar al adversario durante más de 20 segundos o si este se rinde.



Ilustración 10 IPPON EN EJECUCIÓN [1]

- *Waza-ari*, equivale a la mitad del Ippon, se obtiene cuando el oponente es proyectado pero no cae completamente de espaldas. En la Ilustración 11 se puede visualizar la técnica en ejecución por inmovilización.



Ilustración 11 WAZA-ARI POR INMOVILIZACIÓN EN EJECUCIÓN

- *Yuko*, equivale a la mitad de un *Waza-ari* y ocurre cuando el oponente es derribado al suelo y cae de lado.
- *Hansoku-make*, es una acción que significa la descalificación del judoca generalmente al recibir un *Ippon*, o alguna falta que se considere grave.
- *Shido*, es una falta leve y acumulativa, la primera vez suele ser un aviso, la segunda se convierte en un *Yuko* en contra, la tercera un *Waza-ari*, y la 4 un *Hansoku-make*. Es importante agregar que algunas competencias derogan esta norma y solo toman el *Shido* como una falta, sin agregar puntos al adversario en función de la cantidad de faltas, solo cuando llega a 4 (cuatro) *Shido* este se convierte en un *Hansoku-make* y el combatiente es descalificado.

De los eventos descritos anteriormente solo son contados en la tabla de puntuación los *Waza-ari*, *Yuko* y *Shido*, tal como se muestra en la Ilustración 12. Debido que el combate puede ser ganado por *Ippon* o por *Hansoku-make*, entonces al momento de ocurrir estos dos eventos anteriores finaliza el combate.



Ilustración 12 . TABLA DE PUNTUACIÓN

CAPÍTULO 3: MÉTODO DE DESARROLLO

La experiencia de desarrollo de *Software* a través de los años ha indicado que existen problemas recurrentes durante dicho proceso, es por esto, que se han publicado distintos patrones de desarrollo de *Software* con el fin de mejorar la calidad del producto y disminuir los tiempos.

Sommerville [31], define el método de desarrollo de *Software* como “un enfoque estructurado para el desarrollo de *Software* cuyo propósito es facilitar la producción de *Software* de alta calidad de una forma costeable”. Desde 1970 se presentan y publican continuamente diferentes métodos para el desarrollo de *Software*, ninguno es clasificado como ideal, esto depende del área y el personal donde es aplicado.

3.1 MODELO DE PROCESOS DE INGENIERÍA DE SOFTWARE

Los modelos de proceso de ingeniería de *Software* no imponen un modelo concreto de cómo desarrollar, ni como realizar las diferentes actividades incluidas en cada proceso, por lo que cada lector debe instanciar sus propios métodos, técnicas y herramientas.

Según Rosales [32], los modelos de procesos de ingeniería de *Software* se pueden definir como “una simplificación o abstracción de un proceso real”, es decir, consiste en “una representación abstracta de alto nivel de un proceso *Software*”.

Dentro del conjunto de métodos de desarrollo de *Software*, existe una familia de patrones para el desarrollo ágil estos, tiene como objetivo fundamental minimizar las actividades que no se consideran relevantes, aumentar la productividad del equipo de desarrollo y elevar la adaptabilidad del resultado. Por otro lado, hay métodos para el desarrollo de *Software* orientada a prototipos, estos presentan un conjunto de ventajas a los desarrollados, a continuación se presenta una breve descripción de estos [32].

3.1.1 MODELO DE DESARROLLO ÁGIL

Los modelos de desarrollo ágil involucran una filosofía y un conjunto de directrices [33], la filosofía incentiva la satisfacción del cliente y disminuye el tiempo de desarrollo del *Software*, para lograr esta meta se recomienda equipos de desarrollo pequeños y altamente motivados, es ideal minimizar el trabajo en métodos de ingeniería de *Software* y aumentar la simplicidad en el desarrollo del *Software*. Por otro lado, las directrices son aplicadas principalmente en el análisis y diseño, minimizando algunas actividades pero sin descartar su importancia, reforzando esto con una alta y continua comunicación entre el equipo de desarrollo y el cliente.

El proceso de desarrollo ágil representa una alternativa razonable para los métodos de ingeniería tradicional que, normalmente, no se adaptan correctamente a los productos

de *Software* que cambian rápidamente sus requerimientos y evolucionan de forma constante.

Los métodos de desarrollo ágil están basados en pequeñas y rápidas entregas que involucran las actividades de desarrollo, tales como, comunicación, planificación, modelado, construcción y despliegue dentro de un conjunto de tareas mínimas, para disminuir el tiempo entre las entregas.

En cuanto a su característica de entregar el trabajo de *Software* frecuentemente, esto es llamado distinto dependiendo del método específico, Rosales [32] indica que casi todos los autores concuerdan en que el tiempo entre cada entrega no supere las 2 semanas, siempre con preferencia en periodos cortos de tiempo. Por otro lado, es importante que las personas con dominio sobre los procesos del negocio deben trabajar en conjunto con los desarrolladores a lo largo de todo el proyecto.

3.1.2 MODELO BASADO EN PROTOTIPOS

Los modelos de prototipos están enfocados en el especificar los requerimientos presentando al cliente versiones experimentales de un sistema o componente de un sistema que tiene los suficientes elementos como para permitir su utilización, esto son denominados prototipos. Luego de presentar un prototipo son capturados todos los requerimientos solicitados por el cliente, esto con el objetivo de aclarar los requerimientos que el cliente necesita y su percepción del sistema [32].

Inicialmente no se conocen todos los requerimientos del sistema, se generan los prototipos, los clientes prueban y añaden requerimientos, finalmente se realiza una implantación parcial del sistema y se prueba.

Según Rosales [32], las etapas del ciclo de vida del modelo basado en prototipos son:

- Análisis de requisito del sistema
- Análisis de requisitos de *Software*
- Diseño, desarrollo e implementación del prototipo
- Prueba del prototipo
- Refinamiento interactivo del prototipo
- Refinamiento de las especificaciones del prototipo
- Diseño e implementación del sistema final
- Explotación y mantenimiento

Dentro de sus ventajas se encuentran, la alta comunicación entre desarrollador y cliente, estadísticamente se estima que aumenta la posibilidad de aceptación del sistema y es útil cuando el cliente conoce el objetivo general del sistema pero ignora los detalles.

Por otro lado, algunas de sus desventajas son, suelen desatenderse aspectos importantes como la calidad y el mantenimiento, motivado a un desarrollo rápido, el desarrollador suele tomar decisiones de implementación poco convenientes, el cliente espera un sistema completo en lugar de un prototipo y el desarrollador codifica demasiado rápido y motivado a esto ignora muchos detalles.

La construcción o desarrollo basado en prototipos se puede utilizar como un modelo del proceso independiente, comúnmente, se aplica como una técnica implementada dentro del contexto de otro modelo de ingeniería de *Software*. Esto motivado a las ventajas que presenta tanto para el desarrollador del sistema como para el cliente.

3.2 MÉTODO DE DESARROLLO SELECCIONADO

Para el desarrollo de esta investigación fue seleccionado un método de desarrollo ágil, en el cual, la construcción del sistema fue orientado a prototipos. Esto buscando desarrollar un sistema con las ventajas en tiempo y esfuerzo que permiten un desarrollo ágil, mientras se mantiene una fuerte comunicación y relación con el cliente construyendo la solución y recolectando requerimientos utilizando la técnica del modelo de desarrollo orientado a prototipo.

Las fases en el ciclo de vida, como muestra la Ilustración 13, del método de desarrollo seleccionado son:

- Análisis y Diseño
- Desarrollo
- Pruebas
- Implantación

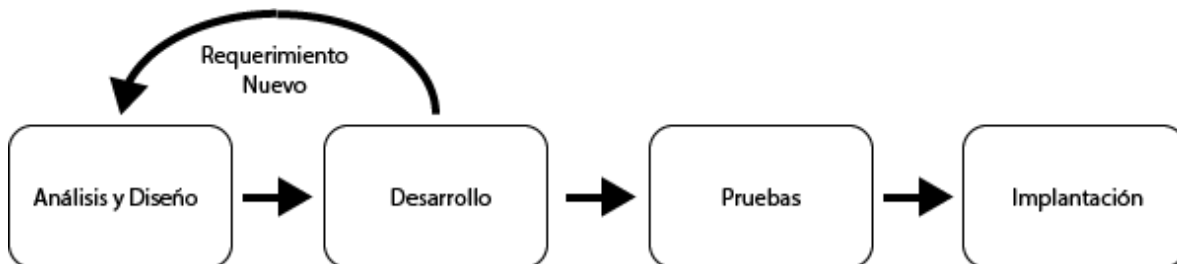


Ilustración 13 FASES DEL MÉTODO DE DESARROLLO

Para la fase de Análisis y Diseño del sistema se seleccionaron las herramientas de diagrama de Casos de Uso y Diagrama de Clases, utilizando el estándar UML. Adicionalmente, cada caso de uso debe tener su descripción.

En el caso de la fase de desarrollo, se seleccionó la técnica del modelo orientado a prototipo, en el cual se capturan los requerimientos y se obtiene retroalimentación del cliente mostrándole un prototipo del sistema. Esto se realiza adaptando los instrumentos de análisis y diseño del mismo, mientras evolucionan los prototipos.

Para la fase de pruebas del sistema, se deben realizar pruebas de aceptación del sistema junto con el cliente en la cual se somete el sistema a una situación, esperando una respuesta acorde, si el sistema no responde de forma correcta deben realizarse los cambios necesarios.

Finalmente, si el cliente está de acuerdo con todas las pruebas de aceptación que le fueron realizadas al sistema, se procede a implantar, instalar el sistema en los dispositivos en los cuales va a ser utilizado por el cliente, configurar todo lo necesario para su despliegue y verificar su correcto funcionamiento.

Adicionalmente, El método de desarrollo seleccionado se basa en la filosofía y las directrices del modelo de desarrollo ágil, adaptadas a la construcción utilizando prototipos. Es decir, utilizando un equipo de desarrollo pequeño, alta comunicación entre las partes involucradas en el proyecto, minimizar algunas actividades de análisis y diseño, periodos cortos de tiempo entre las entregas de prototipos y trabajo en conjunto entre el cliente y los desarrolladores.

CAPÍTULO 4: DESARROLLO DE LA SOLUCIÓN

El desarrollo de la solución se realizó siguiendo las fases del método de desarrollo seleccionado, las cuales son: Análisis y Diseño, Desarrollo, Pruebas e Implementación. En el presente capítulo, se describirán a detalles estas fases, sus procesos y resultados dentro del desarrollo de la solución. Adicionalmente, se explicara la arquitectura de la solución, resaltando su funcionamiento y la conexión entre las aplicaciones y servidores.

4.1 ARQUITECTURA DE LA SOLUCIÓN

Aplicando nuevas tecnologías de recolección y visualización de datos que se adapten a las condiciones actuales de la Selección Venezolana de Judo, se podría mejorar su situación. Motivado esto, se desarrolla una solución de Software para mejorar la captura, almacenamiento y consulta de datos, en la Ilustración 14 se muestra la arquitectura que soportará la solución propuesta en el presente trabajo de investigación.

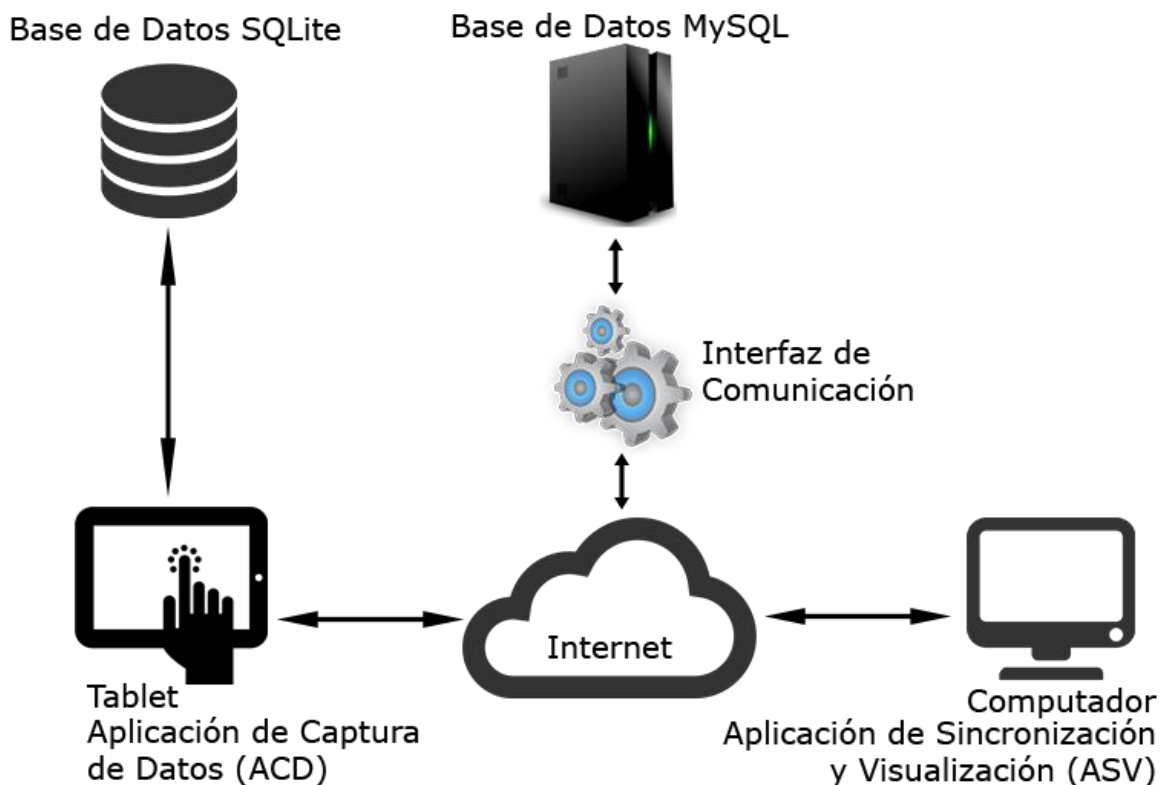


Ilustración 14 ARQUITECTURA DE LA SOLUCIÓN DE SOFTWARE

La propuesta planteada anteriormente consiste en una aplicación móvil desarrollada bajo el sistema operativo *Android*, un servidor de almacenamiento de datos en la nube y una aplicación de escritorio codificada en Java.

La aplicación móvil consistirá en una interfaz de usuario táctil que permitirá mediante botones la rápida captura de eventos durante el combate de judo. Cada botón tendrá asociado un evento característico de esta disciplina y será identificado con el nombre del mismo. Adicionalmente, el usuario tendrá la posibilidad de visualizar el combate mediante la cámara del dispositivo. De esta manera la recolección de los datos se resume en observar el combate por medio de la cámara y pulsar los botones al momento de un evento, sin necesidad de apartar la vista del combate. Es importante agregar que al pulsar un botón se almacena la imagen que la cámara está visualizando en ese instante, además del tiempo que ha transcurrido desde el inicio del combate.

El servidor de almacenamiento de datos permitirá que la información guardada en el dispositivo sea resguardada en un ambiente externo al frágil dispositivo, agregándole beneficios sobre la accesibilidad de los recursos.

La aplicación de escritorio tiene como finalidad la visualización de los datos importados desde el servidor de almacenamiento, utilizando una interfaz de usuario que cumpla los aspectos de usabilidad y experiencia de usuario.

4.2 ANÁLISIS Y DISEÑO DEL SISTEMA

Para este trabajo de investigación se definieron junto con el cliente los requerimientos para la aplicación móvil y la aplicación de escritorio, estos incluyen las funcionalidades necesarias para capturar y consultar los datos de los combates de Judo.

Se utilizó la herramienta de diagrama de Casos de Uso para registrar los requerimientos del cliente, en la Ilustración 15 se muestra los requerimientos de la aplicación móvil utilizando un diagrama de Casos de Uso nivel 1, mientras que, en la Ilustración 16 se pueden visualizar los detalles de estos requerimientos utilizando un diagrama de Casos de Uso nivel 2, por otro lado, en la Ilustración 17 se puede observar los requerimientos de la aplicación de escritorio mediante un diagrama de Casos de Uso nivel 1, y en la Ilustración 18 el diagrama de Casos de Uso nivel 2.

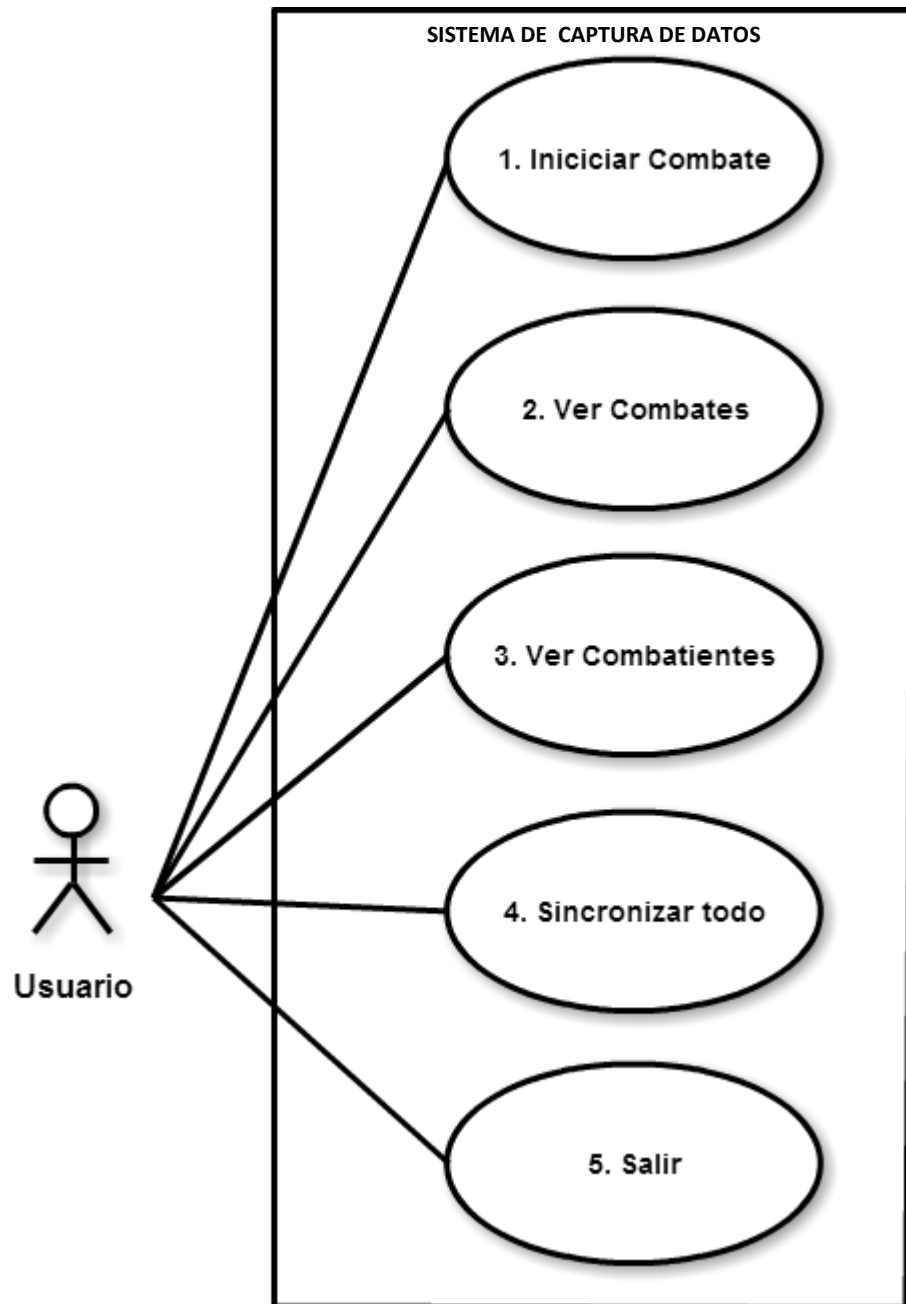


Ilustración 15 DIAGRAMA DE CASOS DE USO NIVEL 1 DE APLICACIÓN MÓVIL

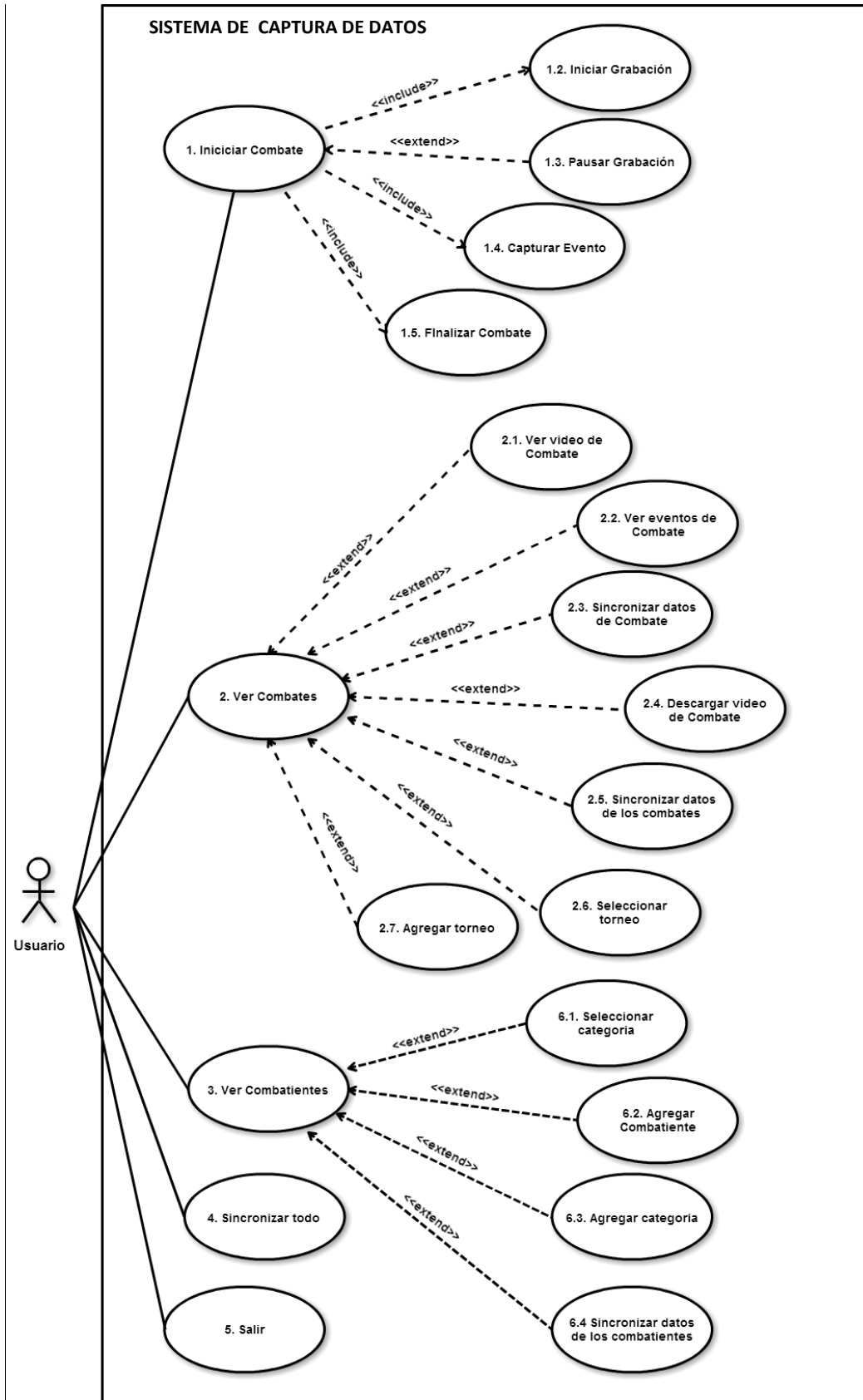


Ilustración 16 DIAGRAMA DE CASOS DE USO NIVEL 2 DE APLICACIÓN MÓVIL

A continuación, se presenta la descripción de los Casos de Uso de la aplicación móvil, cada uno de estos indica la pre-condición que debe cumplir para ser ejecutado, en caso de que aplique, el nombre, los actores involucrados, la descripción del caso de uso, su flujo de funcionamiento básico, el flujo alternativo y la post-condición.

TABLA 1. DESCRIPCIÓN DE CASO DE USO INICIAR COMBATE.

PRE-CONDICIÓN	NINGUNA
NOMBRE	INICIAR COMBATE
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE INICIAR UN COMBATE EN LA APLICACIÓN
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. SE MUESTRA UNA INTERFAZ DONDE SE SOLICITAN LOS DATOS DEL COMBATE: COMBATIENTE AZUL, COMBATIENTE BLANCO Y LUGAR. 2. EL USUARIO INGRESA LOS DATOS. 3. EL SISTEMA CREA EL COMBATE Y SE MUESTRA LA INTERFAZ PRINCIPAL DE CAPTURA DE DATOS DEL COMBATE.
FLUJO ALTERNO	EL USUARIO NO INGRESA LOS DATOS DEL COMBATE Y CANCELA LA INTERFAZ, EL COMBATE NO SE CREA.
POST-CONDICIÓN	EL USUARIO INGRESA A LA INTERFAZ PRINCIPAL DE CAPTURA DE DATOS DEL COMBATE.

TABLA 2. DESCRIPCIÓN DE CASO DE USO INICIAR GRABACIÓN.

PRE-CONDICIÓN	HABER CREADO UN COMBATE
NOMBRE	INICIAR GRABACIÓN
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE INICIAR LA GRABACIÓN DE UN COMBATE EN LA APLICACIÓN
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. SE MUESTRA UNA INTERFAZ PRINCIPAL DE CAPTURA DE DATOS DEL COMBATE. 2. EL USUARIO SELECCIONA INICIAR GRABACIÓN. 3. EL SISTEMA HABILITA LA CAPTURA DE EVENTOS, INICIA EL CRONOMETRO DEL COMBATE Y LA GRABACIÓN DEL VIDEO.
FLUJO ALTERNO	EL USUARIO CANCELA EL INICIO DE COMBATE, EL SISTEMA CIERRA LA INTERFAZ Y RETORNA AL MENÚ PRINCIPAL.
POST-CONDICIÓN	LA GRABACIÓN DEL COMBATE INICIA Y EL USUARIO PUEDE CAPTURAR EVENTOS.

TABLA 3. DESCRIPCIÓN DE CASO DE USO PAUSAR GRABACIÓN.

PRE-CONDICIÓN	HABER INICIADO LA GRABACIÓN DEL COMBATE
NOMBRE	PAUSAR GRABACIÓN
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE PAUSAR LA GRABACIÓN DE UN COMBATE EN LA APLICACIÓN
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. SE MUESTRA UNA INTERFAZ PRINCIPAL DE CAPTURA DE DATOS DEL COMBATE. 2. EL USUARIO SELECCIONA PAUSAR GRABACIÓN. 3. EL SISTEMA DESHABILITA LA CAPTURA DE EVENTOS, PAUSA EL CRONOMETRO DEL COMBATE Y LA GRABACIÓN DEL VIDEO.
FLUJO ALTERNO	EL USUARIO CANCELA LA GRABACIÓN DEL COMBATE, EL SISTEMA CIERRA LA INTERFAZ, EL COMBATE NO ES ALMACENADO Y RETORNA AL MENÚ PRINCIPAL.
POST-CONDICIÓN	LA GRABACIÓN DEL COMBATE Y LA CAPTURA DE EVENTOS SON PAUSADAS.

TABLA 4. DESCRIPCIÓN DE CASO DE USO CAPTURAR EVENTO.

PRE-CONDICIÓN	HABER INICIADO LA GRABACIÓN DEL COMBATE
NOMBRE	CAPTURAR EVENTO
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE REGISTRAR LA OCURRENCIA DE UN EVENTO DURANTE EL COMBATE
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. SE MUESTRA UNA INTERFAZ PRINCIPAL DE CAPTURA DE DATOS DEL COMBATE. 2. EL USUARIO PRESIONA UN BOTÓN DE CAPTURA DE EVENTO. 3. EL SISTEMA REGISTRA EL EVENTO, LO REFLEJA EN LA TABLA DE PUNTAJES DEL COMBATIENTE Y CONTINUA EL COMBATE.
FLUJO ALTERNO	EL USUARIO CANCELA LA CAPTURA, EL EVENTO NO ES REGISTRADO Y CONTINÚA EL COMBATE.
POST-CONDICIÓN	EL SISTEMA REGISTRA EL EVENTO.

TABLA 5. DESCRIPCIÓN DE CASO DE USO FINALIZAR COMBATE.

PRE-CONDICIÓN	HABER INICIADO LA GRABACIÓN DEL COMBATE
NOMBRE	FINALIZAR COMBATE
ACTORES	USUARIO
DESCRIPCIÓN	FINALIZA LA GRABACIÓN DEL COMBATE Y RETORNA AL MENÚ PRINCIPAL DE LA APLICACIÓN.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. SE MUESTRA UNA INTERFAZ PRINCIPAL DE CAPTURA DE DATOS DEL COMBATE. 2. EL USUARIO PRESIONA EL BOTÓN DE FINALIZAR COMBATE. 3. EL SISTEMA FINALIZA LA GRABACIÓN DEL COMBATE Y RETORNA AL MENÚ PRINCIPAL DE LA APLICACIÓN.
FLUJO ALTERNO	EL USUARIO CANCELA LA FINALIZACIÓN Y CONTINUA EL COMBATE.
POST-CONDICIÓN	EL SISTEMA FINALIZA LA GRABACIÓN Y DESPLIEGA LA INTERFAZ DEL MENÚ PRINCIPAL.

TABLA 6. DESCRIPCIÓN DE CASO DE USO VER COMBATES.

PRE-CONDICIÓN	NINGUNA
NOMBRE	VER COMBATES
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE VISUALIZAR UNA LISTA DE LOS COMBATES QUE HAN SIDO CAPTURADOS CON LA APLICACIÓN.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATES CAPTURADOS POR LA APLICACIÓN.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ DE VER COMBATES, EL SISTEMA RETORNA AL MENÚ PRINCIPAL.
POST-CONDICIÓN	EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATES CAPTURADOS POR LA APLICACIÓN.

TABLA 7. DESCRIPCIÓN DE CASO DE USO VER VIDEO DE COMBATE.

PRE-CONDICIÓN	DEBE EXISTIR UN VIDEO EN LA BASE DE DATOS DEL DISPOSITIVO MÓVIL Y DEBE ESTAR ALMACENADO EN EL MISMO.
NOMBRE	VER VIDEO DE COMBATE
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE VISUALIZAR LA GRABACIÓN DE UN COMBATE CAPTURADO CON LA APLICACIÓN.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATES CAPTURADOS POR LA APLICACIÓN. 3. EL USUARIO SELECCIONA UN COMBATE Y PRESIONA EL BOTÓN DE VER VIDEO DEL COMBATE. 4. EL SISTEMA DESPLIEGA LA REPRODUCCIÓN DEL VIDEO DEL COMBATE.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, EL SISTEMA RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA REPRODUCE LA GRABACIÓN DEL COMBATE.

TABLA 8. DESCRIPCIÓN DE CASO DE USO VER EVENTOS DE COMBATE.

PRE-CONDICIÓN	DEBEN EXISTIR EVENTOS DEL COMBATE SELECCIONADO ALMACENADOS EN LA BASE DE DATOS DEL DISPOSITIVO MÓVIL.
NOMBRE	VER EVENTOS DE COMBATE
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE VISUALIZAR LOS EVENTOS DE UN COMBATE CAPTURADO CON LA APLICACIÓN.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATES CAPTURADOS POR LA APLICACIÓN. 3. EL USUARIO SELECCIONA UN COMBATE Y PRESIONA EL BOTÓN DE VER EVENTOS DEL COMBATE. 4. EL SISTEMA DESPLIEGA LOS EVENTOS DEL COMBATE.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, EL SISTEMA RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA DESPLIEGA LOS EVENTOS DEL COMBATE.

TABLA 9. DESCRIPCIÓN DE CASO DE USO SELECCIONAR TORNEO.

PRE-CONDICIÓN	DEBEN EXISTIR TORNEOS EN LA BASE DE DATOS DEL DISPOSITIVO MÓVIL.
NOMBRE	SELECCIONAR TORNEO
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE FILTRAR LOS COMBATES POR TORNEO
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATES CAPTURADOS POR LA APLICACIÓN. 3. EL USUARIO SELECCIONA UN TORNEO. 4. EL SISTEMA DESPLIEGA LOS COMBATES DEL TORNEO SELECCIONADO.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, EL SISTEMA RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA DESPLIEGA LOS COMBATES DEL TORNEO SELECCIONADO.

TABLA 10. DESCRIPCIÓN DE CASO DE USO AGREGAR TORNEO.

PRE-CONDICIÓN	NINGUNA
NOMBRE	AGREGAR TORNEO
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE AGREGAR UN TORNEO
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATES CAPTURADOS POR LA APLICACIÓN. 3. EL USUARIO SELECCIONA AGREGAR TORNEO. 4. EL SISTEMA DESPLIEGA UN FORMULARIO CON LOS DATOS NECESARIOS PARA AGREGAR UN TORNEO. 5. EL USUARIO LLENA LOS CAMPOS REQUERIDOS Y CONFIRMA. 6. EL SISTEMA CREA UN TORNEO CON LOS DATOS SUMINISTRADOS POR EL USUARIO.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, NO ES ALMACENADO EL TORNEO Y EL SISTEMA RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA CREA UN TORNEO CON LOS DATOS SUMINISTRADOS POR EL USUARIO.

TABLA 11. DESCRIPCIÓN DE CASO DE USO SINCRONIZAR DATOS DE COMBATE.

PRE-CONDICIÓN	NINGUNA
NOMBRE	SINCRONIZAR DATOS DE COMBATE
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE EXPORTAR LOS DATOS DEL COMBATE
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATES CAPTURADOS POR LA APLICACIÓN. 3. EL USUARIO SELECCIONA SINCRONIZAR DATOS DE UN COMBATE. 4. EL SISTEMA ENVÍA LOS DATOS DEL COMBATE A LA INTERFAZ DE COMUNICACIÓN.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, NO SE SINCRONIZAN LOS DATOS Y EL SISTEMA RETORNA AL MENÚ PRINCIPAL.
POST-CONDICIÓN	EL SISTEMA ENVÍA LOS DATOS DEL COMBATE A LA INTERFAZ DE COMUNICACIÓN.

TABLA 12. DESCRIPCIÓN DE CASO DE USO SINCRONIZAR DATOS DE LOS COMBATES.

PRE-CONDICIÓN	NINGUNA
NOMBRE	SINCRONIZAR DATOS DE LOS COMBATES
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE HOMOLOGAR LOS DATOS DE LOS COMBATES
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATES CAPTURADOS POR LA APLICACIÓN. 3. EL USUARIO SELECCIONA SINCRONIZAR. 4. EL SISTEMA ENVÍA LOS DATOS DE LOS COMBATES LISTADOS ACTUALMENTE Y RECIBE LA RESPUESTA DE LA INTERFAZ DE COMUNICACIÓN. 5. EL SISTEMA ACTUALIZA LA INFORMACIÓN DE LOS COMBATES.
FLUJO ALTERNO	EL USUARIO CANCELA LA SINCRONIZACIÓN, EL SISTEMA NO SINCRONIZA LOS DATOS Y RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA ENVÍA LOS DATOS DE LOS COMBATES LISTADOS ACTUALMENTE A LA INTERFAZ DE COMUNICACIÓN Y ACTUALIZA LA INFORMACIÓN DE LOS COMBATES.

TABLA 13. DESCRIPCIÓN DE CASO DE USO DESCARGAR VIDEO DE COMBATE.

PRE-CONDICIÓN	EL VIDEO DEL COMBATE DEBE ESTAR EN EL SERVIDOR
NOMBRE	DESCARGAR VIDEO DE COMBATE
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE DESCARGAR EL VIDEO DE UN COMBATE
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATES CAPTURADOS POR LA APLICACIÓN. 3. EL USUARIO SELECCIONA DESCARGAR VIDEO DE COMBATE. 4. EL SISTEMA DESCARGA EL VIDEO DEL COMBATE.
FLUJO ALTERNO	EL USUARIO CANCELA LA DESCARGA, EL SISTEMA NO DESCARGA EL VIDEO Y RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA DESCARGA EL VIDEO DEL COMBATE.

TABLA 14. DESCRIPCIÓN DE CASO DE USO VER COMBATIENTES.

PRE-CONDICIÓN	NINGUNA
NOMBRE	VER COMBATIENTES
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE VISUALIZAR LOS COMBATIENTES ALMACENADOS EN LA APLICACIÓN MÓVIL.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATIENTES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATIENTES ALMACENADOS EN LA APLICACIÓN.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, EL SISTEMA RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATIENTES ALMACENADOS EN LA APLICACIÓN.

TABLA 15. DESCRIPCIÓN DE CASO DE USO SELECCIONAR CATEGORÍA.

PRE-CONDICIÓN	DEBE HABER CATEGORÍAS EN LA BASE DE DATOS DEL DISPOSITIVO MÓVIL.
NOMBRE	SELECCIONAR CATEGORÍA
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE FILTRAR LOS COMBATIENTES ALMACENADOS EN LA APLICACIÓN MÓVIL PARA QUE SEAN VISUALIZADOS.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATIENTES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATIENTES ALMACENADOS EN LA APLICACIÓN. 3. EL USUARIO SELECCIONA UNA CATEGORÍA. 4. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATIENTES DE LA CATEGORÍA SELECCIONADA ALMACENADOS EN LA APLICACIÓN.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, EL SISTEMA RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATIENTES DE LA CATEGORÍA SELECCIONADA ALMACENADOS EN LA APLICACIÓN.

TABLA 16. DESCRIPCIÓN DE CASO DE USO AGREGAR CATEGORÍA.

PRE-CONDICIÓN	NINGUNA
NOMBRE	AGREGAR CATEGORÍA
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE AGREGAR UNA CATEGORÍA A LA BASE DE DATOS DE LA APLICACIÓN MÓVIL.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATIENTES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATIENTES ALMACENADOS EN LA APLICACIÓN. 3. EL USUARIO PRESIONA EL BOTÓN DE AGREGAR CATEGORÍA. 4. EL SISTEMA DESPLIEGA UNA INTERFAZ DE FORMULARIO CON TODOS LOS DATOS NECESARIOS PARA AGREGAR CATEGORÍA. 5. EL USUARIO INGRESA TODOS LOS DATOS DE LA CATEGORÍA Y PRESIONA EL BOTÓN DE GUARDAR. 6. EL SISTEMA AGREGA UNA CATEGORÍA A LA BASE DE DATOS CON LOS DATOS SUMINISTRADOS POR EL USUARIO.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, EL SISTEMA RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA AGREGA UNA CATEGORÍA A LA BASE DE DATOS CON LOS DATOS SUMINISTRADOS POR EL USUARIO.

TABLA 17. DESCRIPCIÓN DE CASO DE USO AGREGAR COMBATIENTE.

PRE-CONDICIÓN	NINGUNA
NOMBRE	AGREGAR COMBATIENTE
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE AGREGAR UN COMBATIENTE A LA BASE DE DATOS DE LA APLICACIÓN MÓVIL.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATIENTES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATIENTES ALMACENADOS EN LA APLICACIÓN. 3. EL USUARIO PRESIONA EL BOTÓN DE AGREGAR COMBATIENTE. 4. EL SISTEMA DESPLIEGA UNA INTERFAZ DE FORMULARIO CON TODOS LOS DATOS NECESARIOS DEL COMBATIENTE. 5. EL USUARIO INGRESA TODOS LOS DATOS DEL COMBATIENTE Y PRESIONA EL BOTÓN DE GUARDAR. 6. EL SISTEMA AGREGA UN COMBATIENTE A LA BASE DE DATOS CON LOS DATOS SUMINISTRADOS POR EL USUARIO.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, EL SISTEMA RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA AGREGA UN COMBATIENTE A LA BASE DE DATOS DE LA APLICACIÓN MÓVIL.

TABLA 18. DESCRIPCIÓN DE CASO DE USO SINCRONIZAR DATOS DE LOS COMBATIENTES.

PRE-CONDICIÓN	NINGUNA
NOMBRE	SINCRONIZAR DATOS DE LOS COMBATIENTES
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE EXPORTAR LOS DATOS DE LOS COMBATIENTES
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATIENTES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATIENTES CAPTURADOS POR LA APLICACIÓN. 3. EL USUARIO SELECCIONA SINCRONIZAR DATOS DE LOS COMBATIENTES. 4. EL SISTEMA ENVÍA LOS DATOS DE LOS COMBATIENTES A LA INTERFAZ DE COMUNICACIÓN.
FLUJO ALTERNO	EL USUARIO CANCELA LA SINCRONIZACIÓN, EL SISTEMA NO SINCRONIZA LOS DATOS Y RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA ENVÍA LOS DATOS DEL COMBATE A LA INTERFAZ DE COMUNICACIÓN.

TABLA 19. DESCRIPCIÓN DE CASO DE USO SINCRONIZAR TODO.

PRE-CONDICIÓN	NINGUNA
NOMBRE	SINCRONIZAR TODO
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE HOMOLOGAR LA BASE DE DATOS DEL DISPOSITIVO CON LA BASE DE DATOS EN LA NUBE
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE SINCRONIZAR TODO. 2. EL SISTEMA BORRA LOS DATOS DE LA BASE DE DATOS. 3. EL SISTEMA ESCRIBE LOS DATOS ENVIADOS POR EL SERVIDOR.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, EL SISTEMA NO SINCRONIZA LOS DATOS Y RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	LA BASE DE DATOS DEL DISPOSITIVO ESTÁ HOMOLOGADA CON LA BASE DE DATOS DEL SERVIDOR.

TABLA 20. DESCRIPCIÓN DE CASO DE USO SALIR.

PRE-CONDICIÓN	NINGUNA
NOMBRE	SALIR
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE CERRAR LA APLICACIÓN
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE SALIR. 2. EL SISTEMA CIERRA LA APLICACIÓN.
FLUJO ALTERNO	NINGUNO
POST-CONDICIÓN	LA APLICACIÓN SE CIERRA.

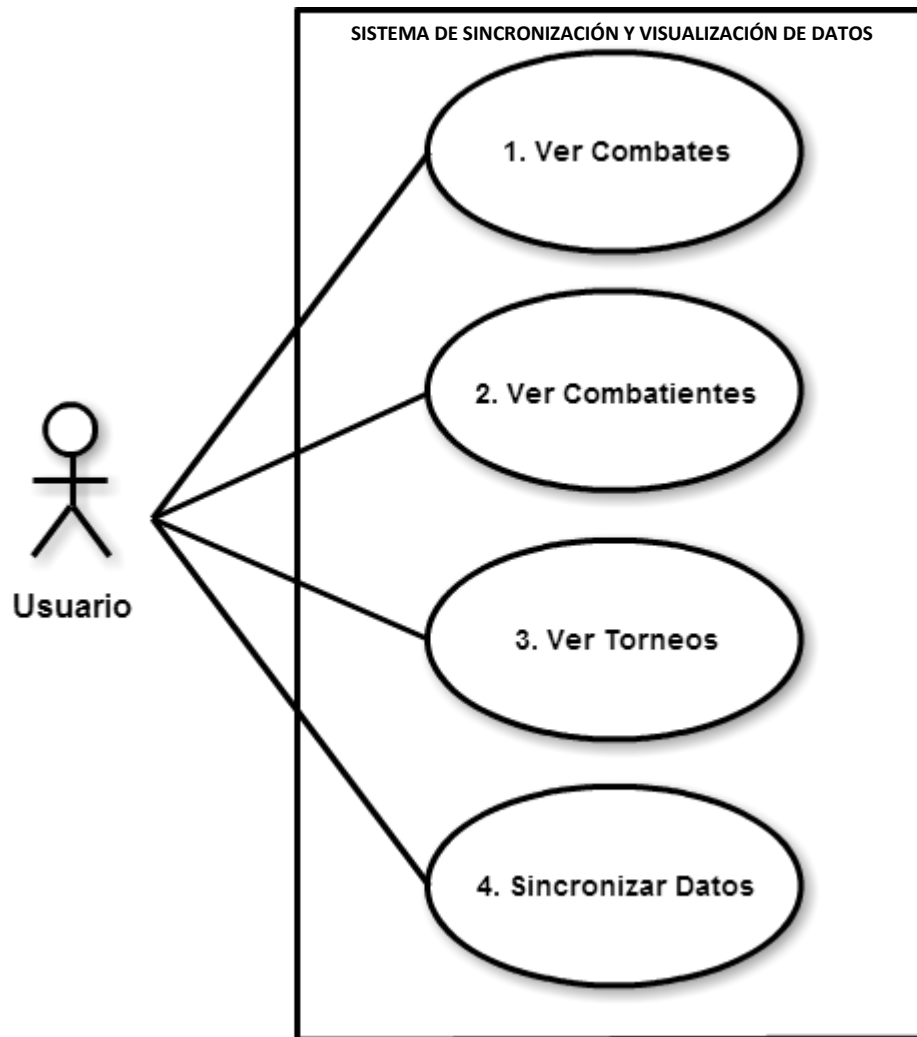


Ilustración 17 DIAGRAMA DE CASOS DE USO NIVEL 1 DE APLICACIÓN DE ESCRITORIO

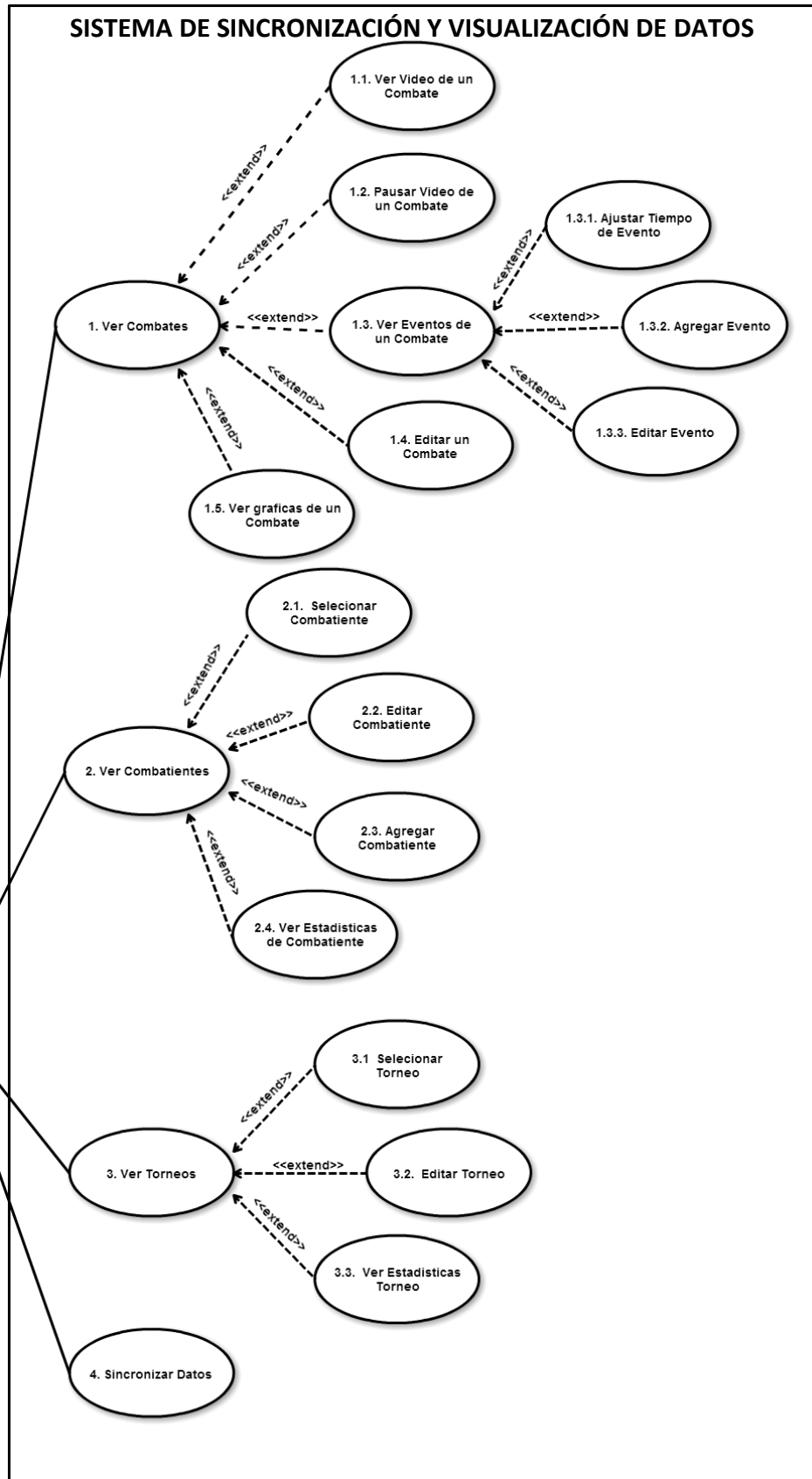


Ilustración 18 DIAGRAMA DE CASOS DE USO NIVEL 2 DE APLICACIÓN DE ESCRITORIO

A continuación, se presenta la descripción de los Casos de Uso de la aplicación de Escritorio, cada uno de estos detalla de la misma manera que los Casos de Uso de la aplicación móvil.

TABLA 21. DESCRIPCIÓN DE CASO DE USO VER COMBATES.

PRE-CONDICIÓN	NINGUNA
NOMBRE	VER COMBATES
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE VER LOS COMBATES ALMACENADOS EN LA APLICACIÓN DE ESCRITORIO.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATES ALMACENADOS EN LA APLICACIÓN.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, EL SISTEMA RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATES ALMACENADOS EN LA APLICACIÓN.

TABLA 22. DESCRIPCIÓN DE CASO DE USO VER VIDEO DE UN COMBATE.

PRE-CONDICIÓN	HABER SELECCIONADO UN COMBATE
NOMBRE	VER VIDEO DE UN COMBATE
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE VER LA GRABACIÓN DE UN COMBATE.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATES ALMACENADOS EN LA APLICACIÓN. 3. EL USUARIO SELECCIONA UN COMBATE Y PRESIONA EL BOTÓN DE VER VIDEO DEL COMBATE. 4. EL SISTEMA REPRODUCE LA GRABACIÓN DEL COMBATE.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, EL SISTEMA RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA REPRODUCE LA GRABACIÓN DEL COMBATE.

TABLA 23. DESCRIPCIÓN DE CASO DE USO SINCRONIZAR DATOS.

PRE-CONDICIÓN	HABER SELECCIONADO UN COMBATE
NOMBRE	SINCRONIZAR DATOS
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE SINCRONIZAR LOS VIDEOS DE LA APLICACIÓN CON EL SERVIDOR QUE ALMACENA LOS VIDEOS DE COMBATES CAPTURADOS CON EL DISPOSITIVO.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE SINCRONIZAR EN EL MENÚ PRINCIPAL. 2. EL SISTEMA SINCRONIZA LOS VIDEOS CON EL SERVIDOR DE ALMACENAMIENTO.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, EL SISTEMA NO SINCRONIZA LOS DATOS Y RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA SINCRONIZA LOS VIDEOS CON EL SERVIDOR DE ALMACENAMIENTO.

TABLA 24. DESCRIPCIÓN DE CASO DE USO VER EVENTOS DE UN COMBATE.

PRE-CONDICIÓN	HABER SELECCIONADO UN COMBATE
NOMBRE	VER EVENTOS DE UN COMBATE
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE VISUALIZAR LOS EVENTOS DE UN COMBATE.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATES ALMACENADOS EN LA APLICACIÓN. 3. EL USUARIO SELECCIONA UN COMBATE Y PRESIONA EL BOTÓN DE VER EVENTOS DEL COMBATE. 4. EL SISTEMA DESPLIEGA UNA INTERFAZ CON UNA LISTA DE LOS EVENTOS CAPTURADOS EN EL COMBATE.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, EL SISTEMA RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA DESPLIEGA UNA INTERFAZ CON UNA LISTA DE LOS EVENTOS CAPTURADOS EN EL COMBATE.

TABLA 25. DESCRIPCIÓN DE CASO DE USO AJUSTAR TIEMPO DE EVENTO.

PRE-CONDICIÓN	HABER SELECCIONADO UN COMBATE
NOMBRE	AJUSTAR TIEMPO DE EVENTO
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE AJUSTAR EL TIEMPO DE UN EVENTO DEL COMBATE.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATES ALMACENADOS EN LA APLICACIÓN. 3. EL USUARIO SELECCIONA UN COMBATE Y PRESIONA EL BOTÓN DE VER EVENTOS DEL COMBATE. 4. EL SISTEMA DESPLIEGA UNA INTERFAZ CON UNA LISTA DE LOS EVENTOS CAPTURADOS EN EL COMBATE. 5. EL USUARIO SELECCIONA UN EVENTO Y PRESIONA EL BOTÓN DE AJUSTAR TIEMPO. 6. EL USUARIO INGRESA EL TIEMPO Y ACEPTA LOS CAMBIOS. 7. EL SISTEMA GUARDA LOS CAMBIOS.
FLUJO ALTERNO	EL USUARIO NO ACEPTA LOS CAMBIOS Y NO SE GUARDA EL TIEMPO MODIFICADO.
POST-CONDICIÓN	EL SISTEMA GUARDA EL EVENTO CON EL TIEMPO MODIFICADO.

TABLA 26. DESCRIPCIÓN DE CASO DE USO EDITAR EVENTO.

PRE-CONDICIÓN	HABER SELECCIONADO UN COMBATE
NOMBRE	EDITAR EVENTO
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE EDITAR UN EVENTO CAPTURADO DURANTE EL COMBATE.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATES ALMACENADOS EN LA APLICACIÓN. 3. EL USUARIO SELECCIONA UN COMBATE Y PRESIONA EL BOTÓN DE VER EVENTOS DEL COMBATE. 4. EL SISTEMA DESPLIEGA UNA INTERFAZ CON UNA LISTA DE LOS EVENTOS CAPTURADOS EN EL COMBATE. 5. EL USUARIO SELECCIONA UN EVENTO Y PRESIONA EL BOTÓN DE AJUSTAR TIEMPO EDITAR 6. EL USUARIO INGRESA LOS DATOS Y ACEPTA LOS CAMBIOS. 7. EL SISTEMA GUARDA LOS CAMBIOS.
FLUJO ALTERNO	EL USUARIO NO ACEPTA LOS CAMBIOS Y NO SE GUARDA EL TIEMPO MODIFICADO.
POST-CONDICIÓN	EL SISTEMA GUARDA EL EVENTO CON LAS MODIFICACIONES.

TABLA 27. DESCRIPCIÓN DE CASO DE USO AGREGAR EVENTO.

PRE-CONDICIÓN	HABER SELECCIONADO UN COMBATE
NOMBRE	AGREGAR EVENTO
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE AGREGAR UN EVENTO AL COMBATE.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATES ALMACENADOS EN LA APLICACIÓN. 3. EL USUARIO SELECCIONA UN COMBATE Y PRESIONA EL BOTÓN DE VER EVENTOS DEL COMBATE. 4. EL SISTEMA DESPLIEGA UNA INTERFAZ CON UNA LISTA DE LOS EVENTOS CAPTURADOS EN EL COMBATE. 5. EL USUARIO PRESIONA EL BOTÓN DE AGREGAR EVENTO. 6. EL USUARIO INGRESA LOS DATOS DEL EVENTO, TIPO DE EVENTO, TIEMPO Y COMPETIDORES AFECTADOS, LUEGO ACEPTA LOS CAMBIOS. 7. EL SISTEMA AGREGA EL EVENTO AL COMBATE.
FLUJO ALTERNO	EL USUARIO NO ACEPTA LOS CAMBIOS Y NO SE AGREGA EL EVENTO AL COMBATE.
POST-CONDICIÓN	EL SISTEMA AGREGA EL EVENTO AL COMBATE.

TABLA 28. DESCRIPCIÓN DE CASO DE USO EDITAR COMBATE.

PRE-CONDICIÓN	HABER SELECCIONADO UN COMBATE
NOMBRE	EDITAR COMBATE
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE EDITAR UN COMBATE CAPTURADO.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATES ALMACENADOS EN LA APLICACIÓN. 3. EL USUARIO SELECCIONA UN COMBATE Y PRESIONA EL BOTÓN DE EDITAR. 4. EL USUARIO INGRESA LOS DATOS Y ACEPTA LOS CAMBIOS. 5. EL SISTEMA GUARDA LOS CAMBIOS.
FLUJO ALTERNO	EL USUARIO NO ACEPTA LOS CAMBIOS Y NO SE GUARDA EL TIEMPO MODIFICADO.
POST-CONDICIÓN	EL SISTEMA GUARDA EL COMBATE LAS MODIFICACIONES.

TABLA 29. DESCRIPCIÓN DE CASO DE USO VER GRÁFICAS DE UN COMBATE.

PRE-CONDICIÓN	HABER SELECCIONADO UN COMBATE
NOMBRE	VER GRÁFICAS DE UN COMBATE
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE VISUALIZAR LAS GRÁFICAS DE UN COMBATE.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATES ALMACENADOS EN LA APLICACIÓN. 3. EL USUARIO SELECCIONA UN COMBATE Y PRESIONA EL BOTÓN DE VER COMBATE. 4. EL SISTEMA DESPLIEGA UNA INTERFAZ CON LOS DATOS EL COMBATE. 5. EL USUARIO PRESIONA EL BOTÓN DE VER GRÁFICAS. 6. EL SISTEMA DESPLIEGA LAS GRÁFICAS DISPONIBLES.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, EL SISTEMA RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA DESPLIEGA LAS GRÁFICAS DISPONIBLES.

TABLA 30. DESCRIPCIÓN DE CASO DE USO VER COMBATIENTES.

PRE-CONDICIÓN	NINGUNA
NOMBRE	VER COMBATIENTES
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE VER LOS COMBATIENTES ALMACENADOS EN LA APLICACIÓN DE ESCRITORIO.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATIENTES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATIENTES ALMACENADOS EN LA APLICACIÓN.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, EL SISTEMA RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATIENTES ALMACENADOS EN LA APLICACIÓN.

TABLA 31. DESCRIPCIÓN DE CASO DE USO SELECCIONAR COMBATIENTE.

PRE-CONDICIÓN	HABER SELECCIONADO UN COMBATIENTE
NOMBRE	SELECCIONAR COMBATIENTE
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE VER LOS DATOS DE UN COMBATIENTE ALMACENADO EN LA APLICACIÓN DE ESCRITORIO.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATIENTES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATIENTES ALMACENADOS EN LA APLICACIÓN. 3. SELECCIONA UN COMBATIENTE Y PRESIONA EL BOTÓN DE VER SUS DATOS. 4. EL SISTEMA DESPLIEGA UNA INTERFAZ CON LOS DATOS DEL COMBATIENTE.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, EL SISTEMA RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS DATOS DEL COMBATIENTE ALMACENADO EN LA APLICACIÓN.

TABLA 32. DESCRIPCIÓN DE CASO DE USO EDITAR COMBATIENTE.

PRE-CONDICIÓN	HABER SELECCIONADO UN COMBATIENTE
NOMBRE	EDITAR COMBATIENTE
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE EDITAR UN COMBATIENTE ALMACENADOS EN LA APLICACIÓN DE ESCRITORIO.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATIENTES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATIENTES ALMACENADOS EN LA APLICACIÓN. 3. EL USUARIO SELECCIONA UN COMBATIENTE Y PRESIONA EL BOTÓN DE EDITAR COMBATIENTE. 4. EL SISTEMA DESPLIEGA UNA INTERFAZ CON LOS DATOS NECESARIOS DEL COMBATIENTE. 5. EL USUARIO INGRESA LOS DATOS DEL COMBATIENTE Y ACEPTA LOS CAMBIOS. 6. EL SISTEMA GUARDA LOS CAMBIOS EN LOS DATOS DEL COMBATIENTE.
FLUJO ALTERNO	EL USUARIO NO ACEPTA LOS CAMBIOS Y NO SE GUARDAN LOS CAMBIOS.
POST-CONDICIÓN	EL SISTEMA GUARDA LOS CAMBIOS EN LOS DATOS DEL COMBATIENTE.

TABLA 33. DESCRIPCIÓN DE CASO DE USO AGREGAR COMBATIENTE.

PRE-CONDICIÓN	NINGUNA
NOMBRE	AGREGAR COMBATIENTE
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE AGREGAR UN COMBATIENTE A LA APLICACIÓN DE ESCRITORIO.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATIENTES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATIENTES ALMACENADOS EN LA APLICACIÓN. 3. EL USUARIO PRESIONA EL BOTÓN DE AGREGAR COMBATIENTE. 4. EL SISTEMA DESPLIEGA UNA INTERFAZ CON LOS DATOS NECESARIOS DEL COMBATIENTE. 5. EL USUARIO INGRESA LOS DATOS DEL COMBATIENTE Y CONFIRMA LOS DATOS. 6. EL SISTEMA AGREGA UN COMBATIENTE CON LOS DATOS SUMINISTRADOS.
FLUJO ALTERNO	EL USUARIO NO CONFIRMA LOS CAMBIOS Y NO SE GUARDAN LOS CAMBIOS
POST-CONDICIÓN	EL SISTEMA AGREGA UN COMBATIENTE CON LOS DATOS SUMINISTRADOS.

TABLA 34. DESCRIPCIÓN DE CASO DE USO VER ESTADÍSTICAS DE UN COMBATIENTE.

PRE-CONDICIÓN	HABER SELECCIONADO UN COMBATIENTE
NOMBRE	VER ESTADÍSTICAS DE UN COMBATIENTE
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE VISUALIZAR LAS GRÁFICAS DE UN COMBATE.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER COMBATIENTES EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS COMBATIENTES ALMACENADOS EN LA APLICACIÓN. 3. EL USUARIO SELECCIONA UN COMBATIENTE Y PRESIONA EL BOTÓN DE VER COMBATIENTE. 4. EL SISTEMA DESPLIEGA UNA INTERFAZ CON LOS DATOS EL COMBATIENTE. 5. EL USUARIO PRESIONA EL BOTÓN DE VER ESTADÍSTICAS. 6. EL SISTEMA DESPLIEGA LAS ESTADÍSTICAS DISPONIBLES.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, EL SISTEMA RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA DESPLIEGA LAS ESTADÍSTICAS DISPONIBLES.

TABLA 35. DESCRIPCIÓN DE CASO DE USO VER TORNEOS.

PRE-CONDICIÓN	NINGUNA
NOMBRE	VER TORNEOS
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE VER LOS TORNEOS ALMACENADOS EN LA APLICACIÓN DE ESCRITORIO.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER TORNEOS EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS TORNEOS ALMACENADOS EN LA APLICACIÓN.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, EL SISTEMA RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS TORNEOS ALMACENADOS EN LA APLICACIÓN.

TABLA 36. DESCRIPCIÓN DE CASO DE USO SELECCIONAR TORNEO.

PRE-CONDICIÓN	EXISTIR TORNEOS EN BASE DE DATOS
NOMBRE	SELECCIONAR TORNEO
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE VER LOS DATOS DE UN TORNEO ALMACENADO EN LA APLICACIÓN DE ESCRITORIO.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER TORNEOS EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS TORNEOS ALMACENADOS EN LA APLICACIÓN. 3. SELECCIONA UN TORNEO Y PRESIONA EL BOTÓN DE VER TORNEO. 4. EL SISTEMA DESPLIEGA UNA INTERFAZ CON LOS DATOS DEL TORNEO.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, EL SISTEMA RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS DATOS DEL TORNEO ALMACENADO EN LA APLICACIÓN.

TABLA 37. DESCRIPCIÓN DE CASO DE USO EDITAR TORNEO.

PRE-CONDICIÓN	HABER SELECCIONADO UN TORNEO
NOMBRE	EDITAR TORNEO
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE EDITAR UN TORNEO ALMACENADOS EN LA APLICACIÓN DE ESCRITORIO.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER TORNEOS EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS TORNEOS ALMACENADOS EN LA APLICACIÓN. 3. EL USUARIO SELECCIONA UN TORNEO Y PRESIONA EL BOTÓN DE EDITAR TORNEO. 4. EL SISTEMA DESPLIEGA UNA INTERFAZ CON LOS DATOS NECESARIOS DEL TORNEO. 5. EL USUARIO INGRESA LOS DATOS DEL TORNEO Y ACEPTA LOS CAMBIOS. 6. EL SISTEMA GUARDA LOS CAMBIOS EN LOS DATOS DEL TORNEO.
FLUJO ALTERNO	EL USUARIO NO ACEPTA LOS CAMBIOS Y NO SE GUARDAN LOS CAMBIOS.
POST-CONDICIÓN	EL SISTEMA GUARDA LOS CAMBIOS EN LOS DATOS DEL TORNEO.

TABLA 38. DESCRIPCIÓN DE CASO DE USO VER ESTADÍSTICAS DE UN TORNEO.

PRE-CONDICIÓN	HABER SELECCIONADO UN TORNEO
NOMBRE	VER ESTADÍSTICAS DE UN TORNEO
ACTORES	USUARIO
DESCRIPCIÓN	PERMITE VISUALIZAR LAS GRÁFICAS DE UN TORNEO.
FLUJO BÁSICO	<ol style="list-style-type: none"> 1. EL USUARIO PRESIONA EL BOTÓN DE VER TORNEO EN EL MENÚ PRINCIPAL. 2. EL SISTEMA DESPLIEGA UNA INTERFAZ CON TODOS LOS TORNEOS ALMACENADOS EN LA APLICACIÓN. 3. EL USUARIO SELECCIONA UN TORNEO Y PRESIONA EL BOTÓN DE VER TORNEO. 4. EL SISTEMA DESPLIEGA UNA INTERFAZ CON LOS DATOS EL TORNEO. 5. EL USUARIO PRESIONA EL BOTÓN DE VER ESTADÍSTICAS. 6. EL SISTEMA DESPLIEGA LAS ESTADÍSTICAS DISPONIBLES.
FLUJO ALTERNO	EL USUARIO CIERRA LA INTERFAZ, EL SISTEMA RETORNA A LA INTERFAZ ANTERIOR.
POST-CONDICIÓN	EL SISTEMA DESPLIEGA LAS ESTADÍSTICAS DISPONIBLES.

A continuación se describe la estructura del sistema, utilizando un diagrama de clases que puede ser visualizado en la Ilustración 19, en el cual se describen las clases que conforman el sistema, sus atributos, métodos y relaciones, tal y como fueron identificados y modelados según los requerimientos del cliente, utilizando el lenguaje UML.

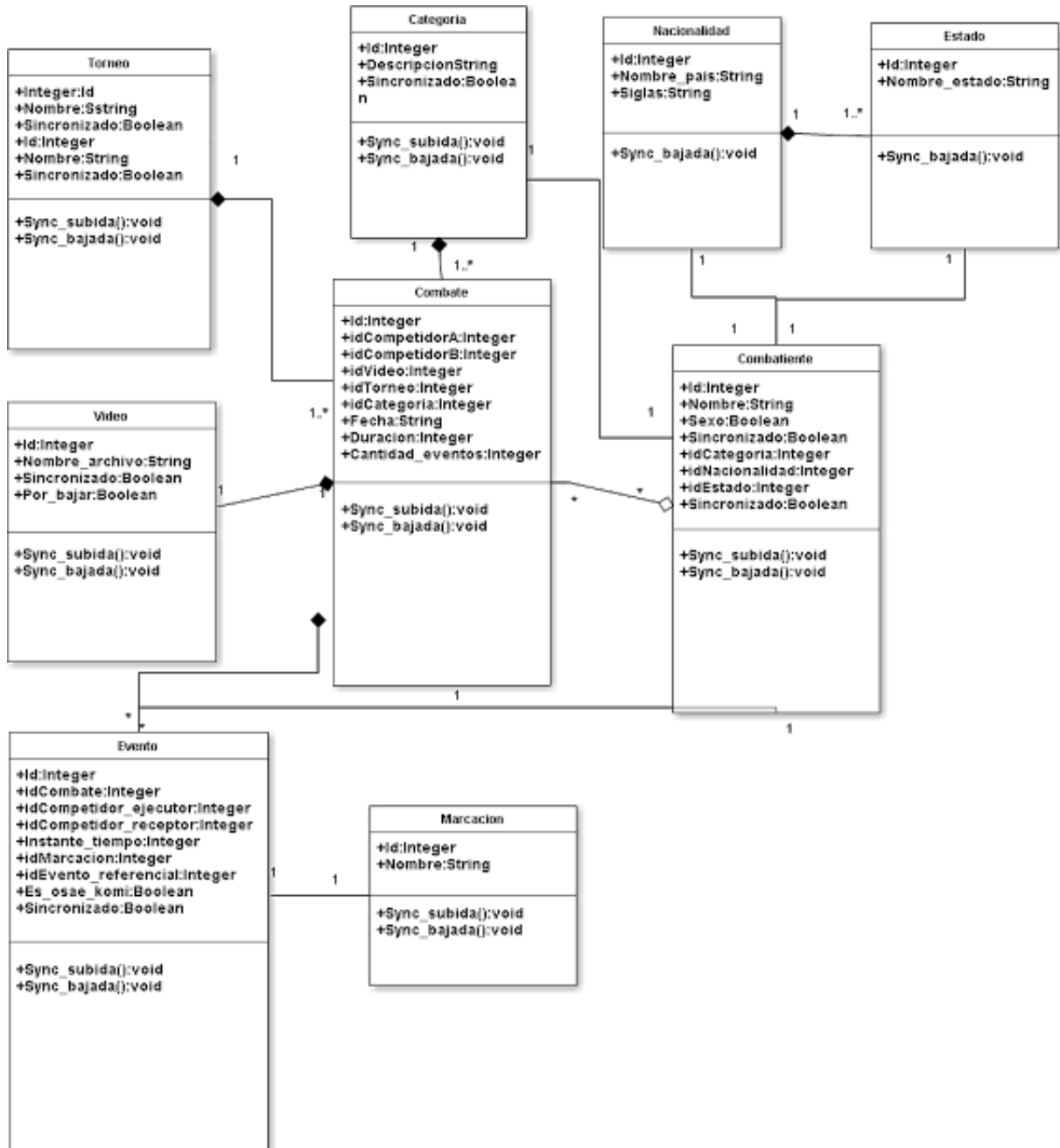


Ilustración 19 DIAGRAMA DE CLASES DEL SISTEMA

La Ilustración 20 corresponde al modelo de base de datos de SQLite, que funciona para almacenar temporalmente los datos en el dispositivo móvil para que luego sean enviados a la base de datos MySQL utilizando la interfaz de comunicación web.

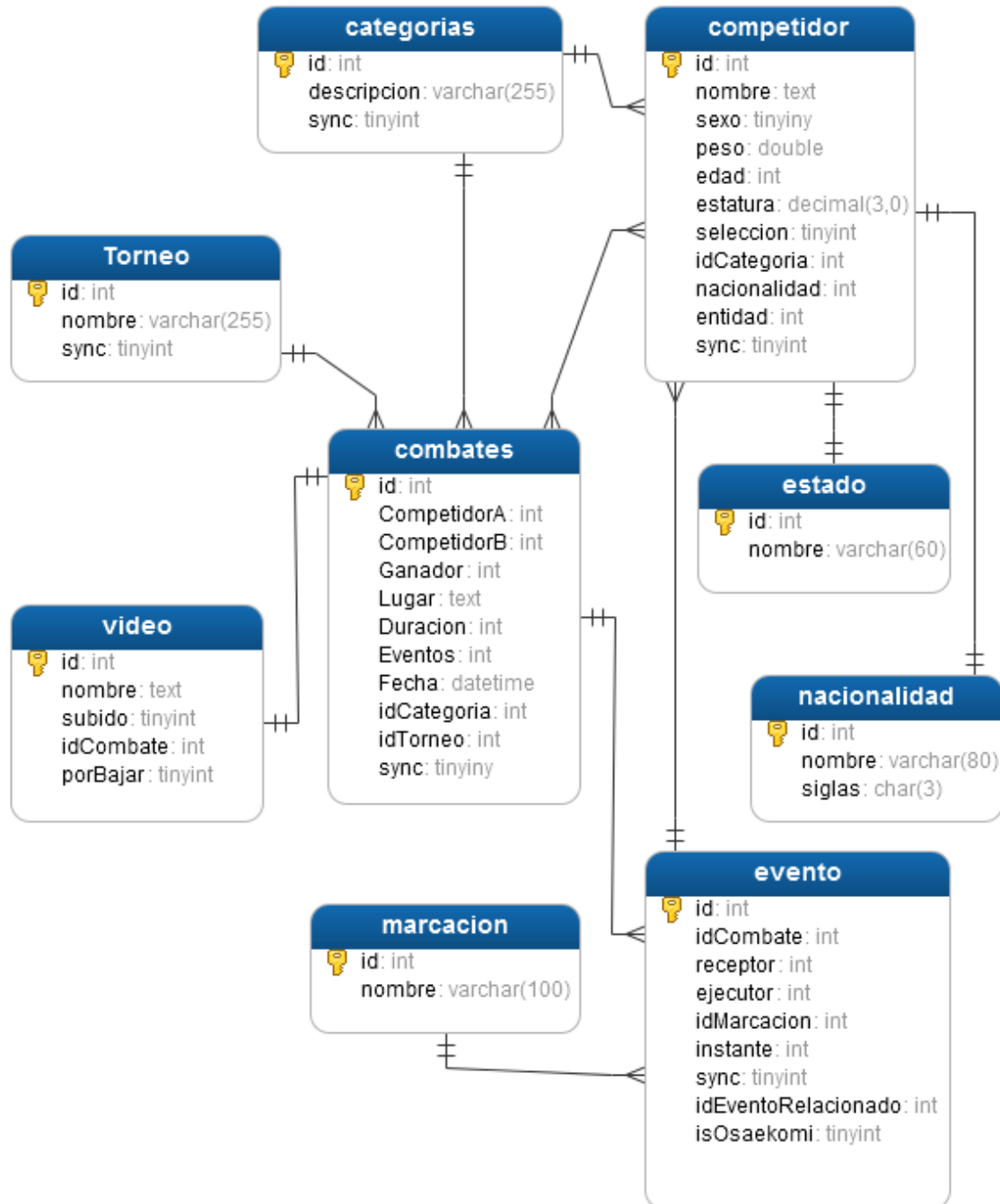


Ilustración 20 MODELO DE BASE DE DATOS SQLITE DE APLICACIÓN MÓVIL

La Ilustración 21 corresponde al modelo de base de datos de MySQL, donde son almacenados los datos enviados por el dispositivo móvil, que son escritos por la interfaz de comunicación web y visualizados por la aplicación de escritorio.

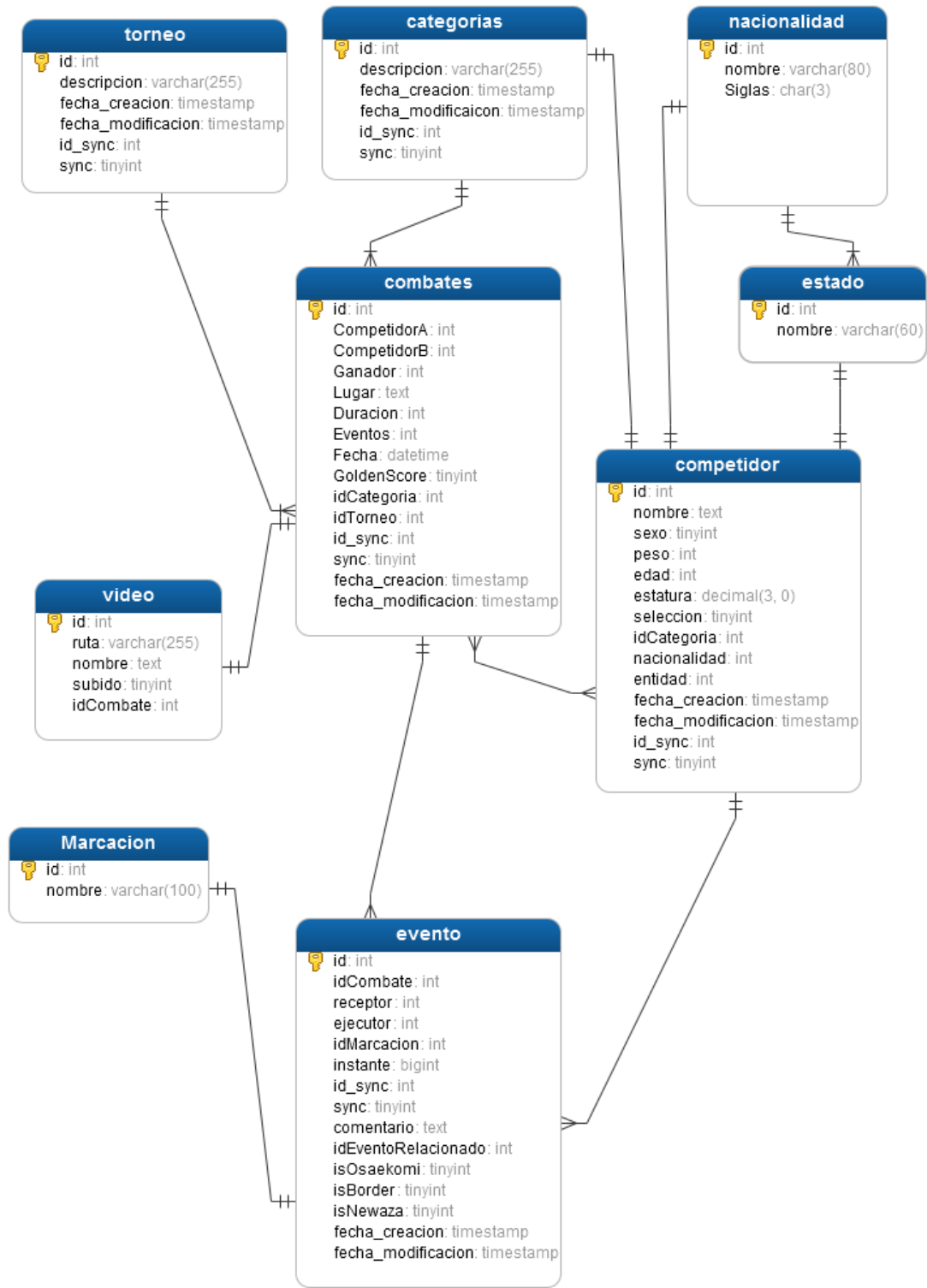


Ilustración 21 MODELO DE BASE DE DATOS MYSQL DE APLICACIÓN DE ESCRITORIO

4.3 DESARROLLO

Para el desarrollo de la solución de este trabajo de investigación se decidió utilizar el IDE *Android Studio*, motivado a su estabilidad, prestaciones para pruebas y depuración de código, por otro lado, se utilizó el IDE *Netbeans*, por sus prestaciones para el diseño de interfaces utilizando el lenguaje de programación Java. Se diseñaron las interfaces móviles aplicando mejores prácticas de usabilidad en interfaces de bajas dimensiones y táctiles, de forma similar, se tomaron en cuenta mejores prácticas en el desarrollo de *Software* para computadores de escritorio.

4.3.1 DESARROLLO DE APLICACIÓN MÓVIL

Para el desarrollo de la aplicación para dispositivo móvil bajo la plataforma Android se necesitó desarrollar tres (3) prototipos y la aplicación final. Cada prototipo desarrollado fue mostrado al cliente, este solicitó los cambios requeridos, tanto en funcionalidad como en diseño. Con esto se construyó una idea sólida de los requerimientos del cliente, buscando una solución final de calidad.

4.3.1.1 PROTOTIPO INICIAL

En reuniones previas con el cliente se capturaron los requerimientos generales de la aplicación, se tomaron las medidas técnicas para su desarrollo y se probaron las funcionalidades en repetidas ocasiones antes de su presentación.

Para el prototipo inicial, se desarrolló la sincronización de los combates, la cual se realizaba todos al mismo tiempo, adicionalmente se incluyó la funcionalidad que permite capturar un combate, tanto un video de la pelea como de los eventos que sucedieron en él, los cuales pueden ser: *Ippon*, *Waza-ari*, *Juko*, *Shido* y *Hansoku-make*. Se agregó un reloj en la interfaz del combate que permite manejar el tiempo y funcionar como guía para el usuario, además, dicho reloj funciona también para pausar y reanudar la grabación del combate. Finalmente, el prototipo permite almacenar el video y los datos de los eventos del combate en el dispositivo móvil.

En la Ilustración 22 podemos visualizar la interfaz de grabación de combate del prototipo inicial en un teléfono con pantalla de 4,7 pulgadas, la cual incluye las funcionalidades de captura mencionadas anteriormente.

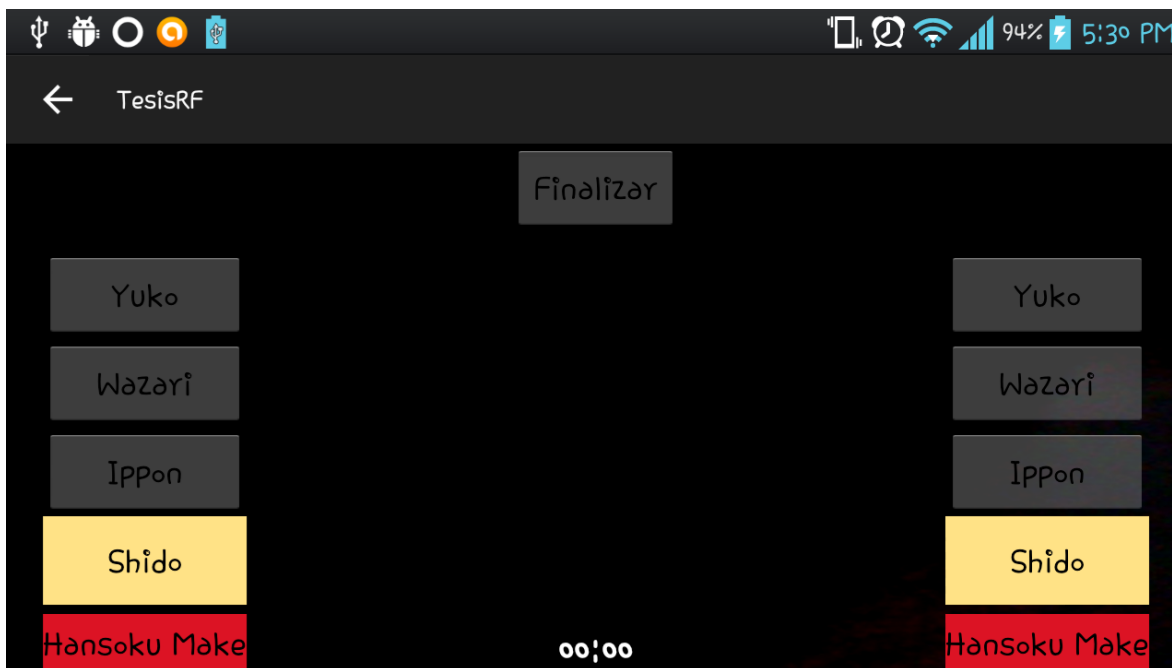


Ilustración 22 INTERFAZ DE GRABACIÓN DE COMBATE DEL PROTOTIPO INICIAL

4.3.1.2 REVISIONES DEL PROTOTIPO INICIAL POR PARTE DEL CLIENTE

Luego de presentar el prototipo inicial al cliente, surgieron un conjunto de requerimientos y modificaciones necesarias, motivado a la necesidad de evaluar cuando ocurre que un combatiente ataca otro pero no logra una marcación, el cliente solicitó agregar la posibilidad de capturar el evento de ataque, por otro lado, con el objetivo de conocer cuál es el estado actual del combate, el cliente solicitó agregar la tabla de marcaciones, donde se presentan la cantidad de *Yuko*, *Waza-ari* y *Shido* que el combatiente ha marcado durante el combate.

Motivado a que el reloj que se encuentra en la interfaz de grabación de combate funciona como un botón para pausar y reanudar la captura del combate, el cliente solicitó un estilo más similar a un botón, con el objetivo de reconocer con facilidad dicha funcionalidad.

Por otro lado, el cliente solicitó la capacidad de ver en el dispositivo los combates y los combatientes que están almacenados en el dispositivo. Además, al seleccionar un combate específico permitir visualizar los eventos que sucedieron en ese combate y reproducir el video del combate.

Para el cliente también fue necesario tener la capacidad de crear combatientes desde el dispositivo móvil para que participen en un combate, finalmente, permitir sincronizar los combates uno por uno y bajo demanda.

4.3.1.3 SEGUNDO PROTOTIPO

Para el segundo prototipo se agregaron las funcionalidades solicitadas por el cliente en la presentación del prototipo inicial, tales como, agregar el evento “Ataque” en la captura de combate, el cual permite capturar un evento que no genere marcación, en la misma interfaz se agregó la tabla de puntuación de cada combatiente y la capacidad de borrar el último evento capturado, también se complementó el reloj con un estilo de botón.

En la funcionalidad general de la aplicación, se agregó la capacidad de ver los combates capturados y los combatientes almacenados en el dispositivo, además de la capacidad de crear combatientes, ver los eventos y el video de un combate seleccionado.

En la Ilustración 23 podemos visualizar la interfaz de grabación de combate del segundo prototipo en un teléfono con pantalla de 4,7 pulgadas, se puede apreciar los cambios producto de las solicitudes del cliente, incluye las funcionalidades mencionadas en el prototipo inicial sumándole las modificaciones solicitadas.

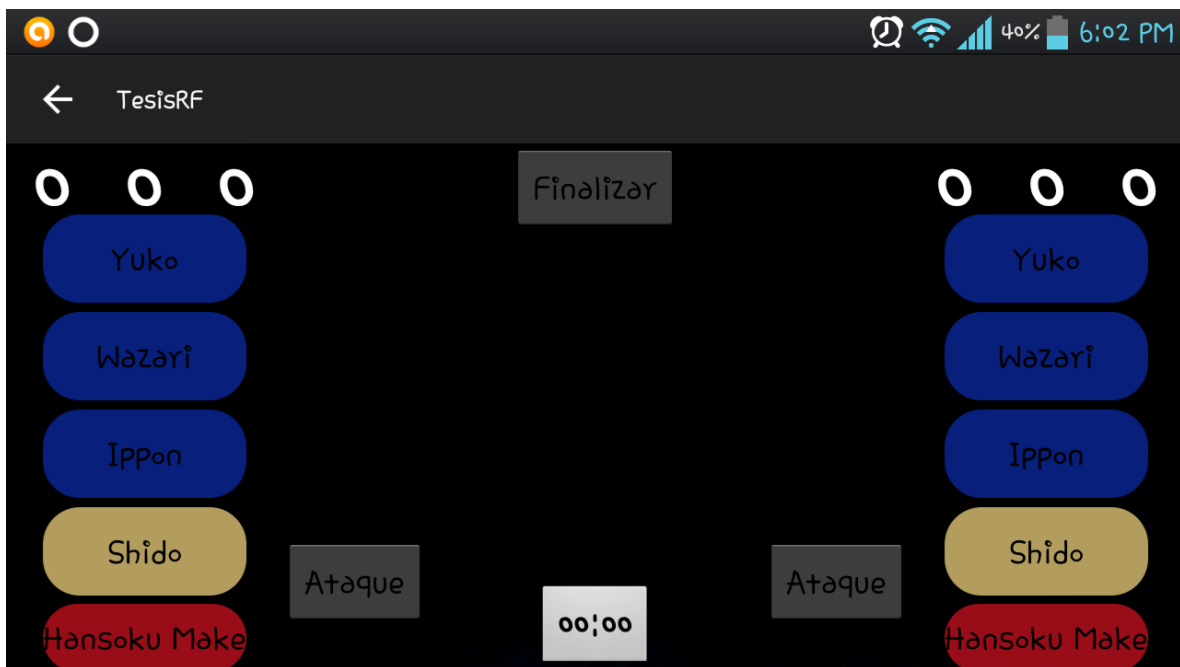


Ilustración 23 INTERFAZ DE GRABACIÓN DE COMBATE DEL SEGUNDO PROTOTIPO

4.3.1.4 REVISIONES DEL SEGUNDO PROTOTIPO POR PARTE DEL CLIENTE

En este prototipo se obtuvo por parte del cliente la solicitud de cambiar el orden y color de los botones de la interfaz de “Grabar combate” esto con el fin de ubicar aquellos con los que por naturaleza se interactúa mayor cantidad de veces más accesibles y los colores para relacionarlos con cada competidor durante el combate. Además, se solicitó la creación de “Categorías” para los combatientes dentro del sistema. En cuanto a las marcaciones, por parte de los miembros del dojo de judo UCV surgió la necesidad de agregar un nuevo tipo de marcación “osae-komi” que deberá ser implementada en el próximo prototipo de esta aplicación. Adicionalmente de lo anterior se sugirió agregar iconos con metáforas a las funcionalidades de “ver combate” “sincronizar combate” “ver eventos de combate”.

4.3.1.5 TERCER PROTOTIPO

En el tercer prototipo de la aplicación se incorporó un conjunto de funcionalidades referentes a las marcaciones durante los combates, la más importante es la marcación “osae-komi” que puede ser activada cuando se produce una inmovilización, está activa un cronometro que le sirve al usuario como referencia en caso de que esta técnica termine en marcación, además de eso , cada conjunto de botones perteneciente a un competidor fue teñido con los colores del mismo y además estos fueron reordenados. En cuanto a los “tableros” de marcaciones, estos se adaptaron visualmente para coincidir con los que se usan en los combates reales, esto con el fin de facilitar la interacción con la aplicación en esta interfaz. En lo relacionado con los combatientes, se agregó la funcionalidad de crear, asignar, y filtrar por categoría. Finalmente se revisó la sincronización en general agregando los nuevos parámetros y se acomodaron algunos errores que se presentaban en prototipos anteriores.

En la Ilustración 24 podemos visualizar la interfaz de grabación de combate del tercer prototipo en el formato de Tablet con una pantalla de 7 pulgadas y sistema operativo *Android* 4.4.4 Kit-Kat que el IDE *Android Studio* permite emular.

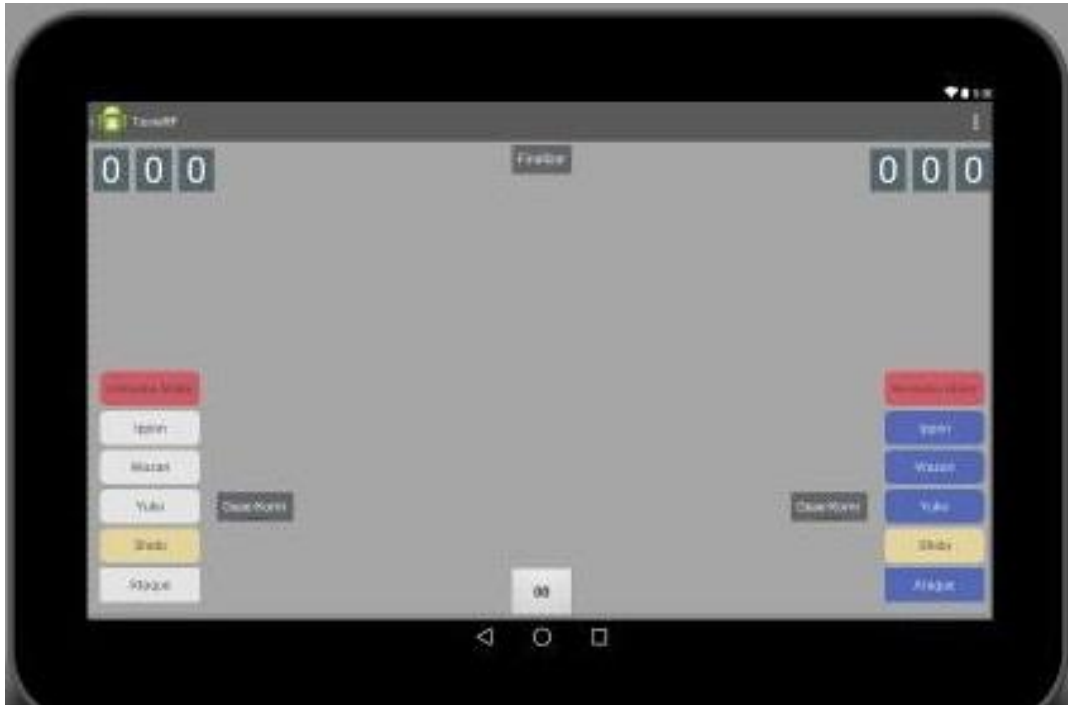


Ilustración 24 INTERFAZ DE GRABACIÓN DE COMBATE DEL TERCER PROTOTIPO

4.3.1.6 REVISIONES DEL TERCER PROTOTIPO POR PARTE DEL CLIENTE

Luego de la presentación de este prototipo surgió por parte del cliente la inquietud en cuanto a la creación de “Torneos” dentro del sistema. En cuanto a los combates se solicitó que pudiesen asignarse a un torneo y una categoría además, en caso de que el video del combate sea eliminado o se haya extraviado del dispositivo debe existir la posibilidad de descargarlo desde el servidor externo, funcionalidad a tomar en cuenta en el siguiente modelo. Adicionalmente deberá poder asignarse una nacionalidad a cada competidor y en el caso de ser venezolana una entidad. Finalmente se deben revisar errores en la sincronización y dar la posibilidad de sincronizar todo lo que está en la base de datos externa, esto en caso de adquirir un nuevo dispositivo o simplemente porque así se desee.

4.3.1.7 PROTOTIPO FINAL

Para la solución de *Software* final, cuya interfaz de grabar combate se puede ver en la Ilustración 25, se codificaron todos los requerimientos de prototipos anteriores, se agregaron parámetros de calidad de *Software* y las revisiones que dio el cliente. Esto último incluye la capacidad de crear, listar y filtrar torneos, además, de permitir la descarga de un video, en caso de que por alguna razón haya sufrido pérdidas del contenido capturado.

Por otro lado, también se le asignó una categoría a un combate y, en cuanto a la interfaz de combate, se modificó el texto del botón de inicio de grabación de combate a

iniciar y se homologaron los colores y transparencias de los botones. Además, la aplicación permite establecer una nacionalidad y una entidad al momento de crear un combatiente.

Finalmente, en caso de que se necesite cambiar un dispositivo móvil, la aplicación se le agrego la funcionalidad de sincronizar toda la base de datos con los datos que se encuentran en el servidor externo, mediante la interfaz de comunicación web.

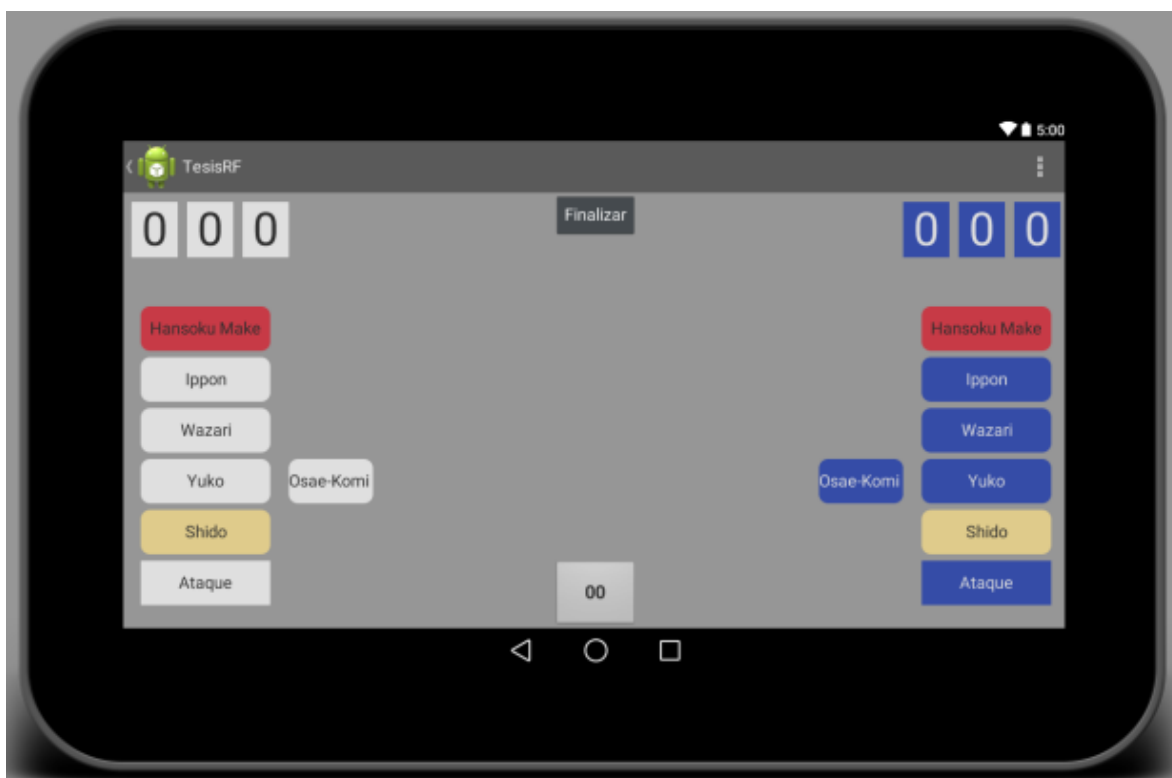


Ilustración 25 INTERFAZ DE COMBATE DEL PROTOTIPO FINAL DE LA APLICACIÓN MÓVIL

4.3.2 DESARROLLO DE APLICACIÓN DE ESCRITORIO

Durante el desarrollo de la aplicación de escritorio se presentó un prototipo y la versión final del aplicativo. Esto, debido a la experiencia del negocio producto del desarrollo de la aplicación bajo la plataforma *Android*.

4.3.2.1 PROTOTIPO INICIAL

El prototipo inicial cumplía las funcionalidades que se habían discutido en reuniones previas con el cliente, la Ilustración 26 muestra la interfaz principal de la aplicación, entre las funcionalidades incluidas están, la capacidad de ver todos los combates y combatientes que han sido capturados con la aplicación móvil, editarlos y ver en detalle sus datos, específicamente, la capacidad de ver gráficas de los combates y los combatientes. Lo

explicado anteriormente fue posible desarrollarlo, producto que, los eventos asociados a cada combate y la sincronización realizada desde el dispositivo a la aplicación de escritorio.



Ilustración 26 INTERFAZ PRINCIPAL DEL PROTOTIPO INICIAL DE LA APLICACIÓN DE ESCRITORIO

En el caso de los combatientes, la aplicación permite crear combatientes que luego serán reflejados en la aplicación móvil, ver sus datos detalladamente, tales como, sexo, edad, peso y si pertenece o no a la selección Venezolana. Además, entre las gráficas que pueden visualizarse de los combatientes están: Marcaciones, Combates, Frecuencia de ataque, Perfil general, Estilo de marcación, Resultados en tiempo reglamentario y Resultados en *golden score*.

La gráfica de marcaciones, tal como se puede observar en la Ilustración 27, refleja las marcaciones que un combatiente ha realizado bien sea: Ataque, *Ippon*, *Waza-ari*, *Juko*, *Shido* o *Hansoku-make*.



Ilustración 27 GRÁFICA DE MARCACIONES

El indicador de combates, muestra la relación del conjunto de combates ganados y perdidos, agregando la marcación mediante la cual se decidió el ganador del combate, es decir, bien sea que el combatiente haya perdido o ganado el combate, se indica si este fue por *Ippon*, *Waza-ari*, *Juko*, *Shido* o *Hansoku-make*. Esto se puede apreciar en la Ilustración 28.

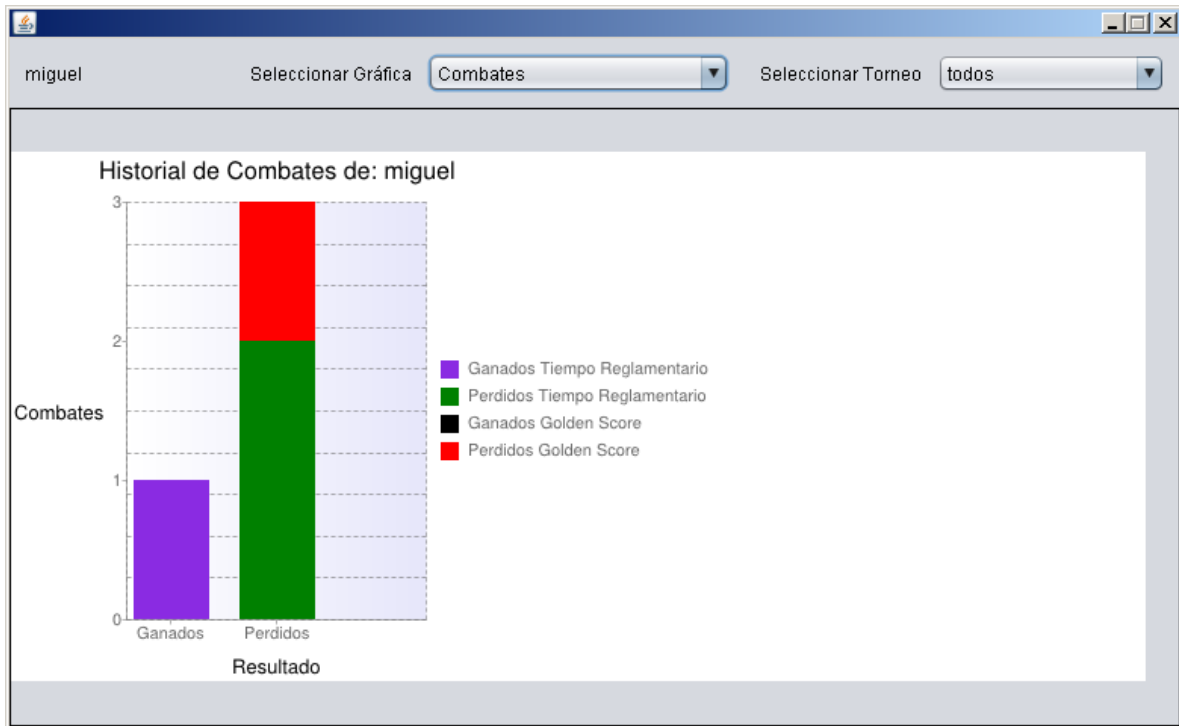


Ilustración 28 GRÁFICA DE COMBATES

La línea de tiempo de frecuencia de ataque, como se puede observar en la Ilustración 29, refleja la cantidad de ataques que fueron realizados en el tiempo de combate, durante todos los combates que haya realizado dicho competidor.

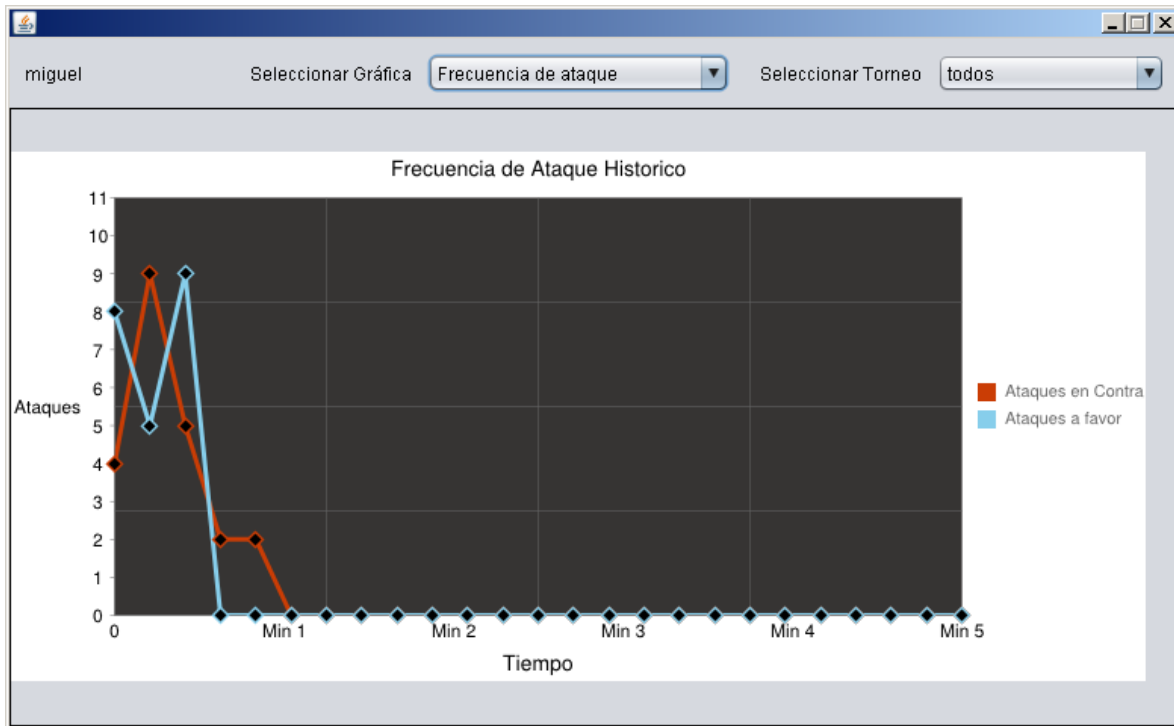


Ilustración 29 GRÁFICA DE FRECUENCIA DE ATAQUE

La gráfica de perfil general es un diagrama de radar conformado por un conjunto de dianas, como refleja la Ilustración 30, su concepción expresa que mientras más alejado del centro del gráfico se encuentre un punto mejor será el indicador al que está referido. En el caso del diagrama que aporta la aplicación de escritorio está compuesto por tres (3) indicadores, el primero, está relacionado con el tiempo que ha transcurrido entre cada ataque generado por un competidor, esto se logra calcular realizando un cociente entre la sumatoria de todos los ataques realizados y el tiempo transcurrido en cada combate, el segundo, refleja la tasa de precisión, es decir, la relación entre el conjunto de ataques del combatiente que generaron una marcación y el conjunto de ataques totales que ha realizado, finalmente, el último indicador, expresa la tasa de ataque entre la defensa del combatiente, es decir, la relación matemática entre la cantidad de ataques realizados y el número de ataques recibidos durante todos los combates que haya realizado el competidor.

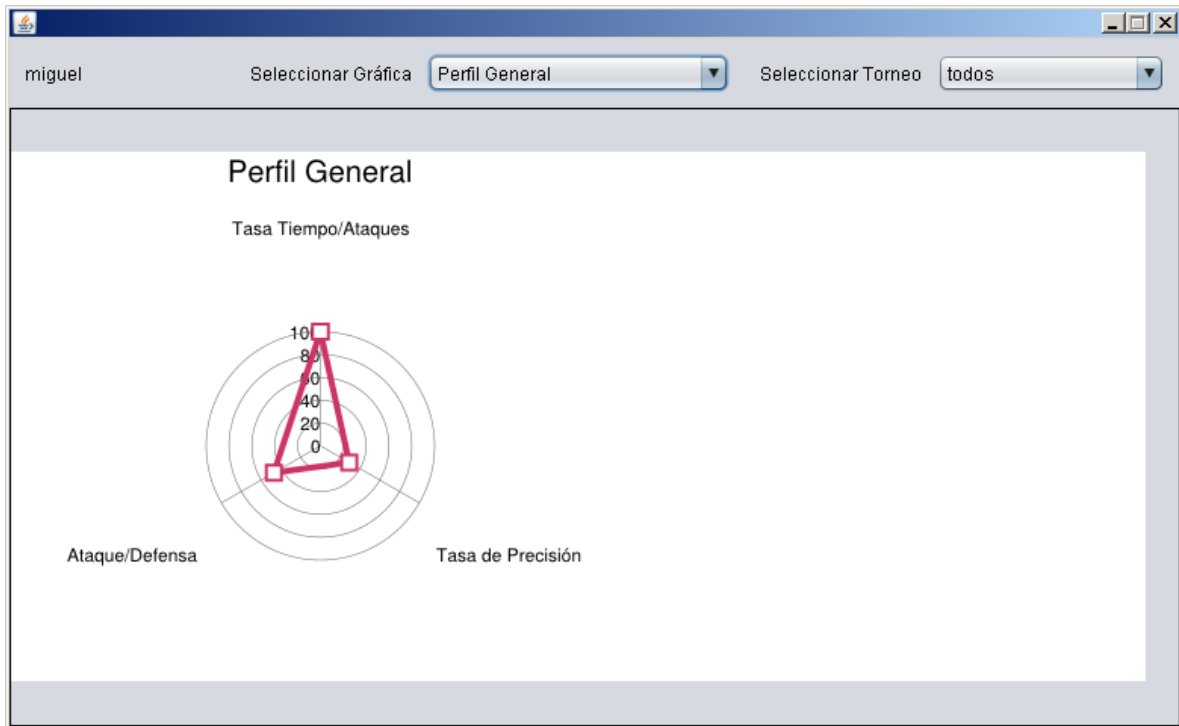


Ilustración 30 GRÁFICA DE PERFIL GENERAL

En cuanto a la gráfica de estilo de marcación, esta permite al usuario visualizar la manera en la cual el combatiente ha realizado sus marcaciones, es decir, si ha sido en el borde del área de combate, en el piso, denominado *Newaza*, por inmovilización, denominado *Osa-komi* u otro estilo de marcación que no está clasificado dentro de las anteriores. Se puede apreciar un ejemplo en la Ilustración 31.

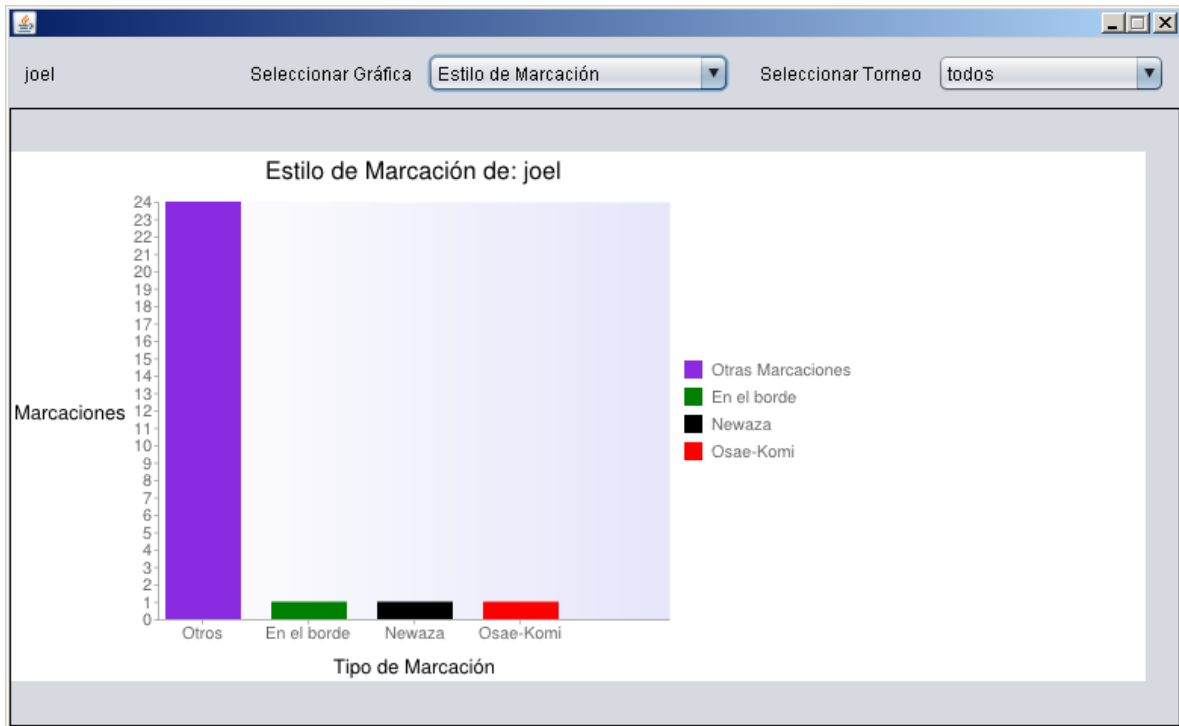


Ilustración 31 GRÁFICA DE ESTILO DE MARCACIÓN

Finalmente, las gráficas de resultados en tiempo *golden score*, cuyo ejemplo se puede ver en la Ilustración 32, y resultados en tiempo reglamentario, que se muestra en la Ilustración 33, indican la relación del conjunto de combates ganados y perdidos, tanto en tiempo reglamentario como en *golden score*, mostrando adicionalmente la técnica mediante la cual se decidió el ganador del combate, al igual que en el gráfico de indicador de combates.

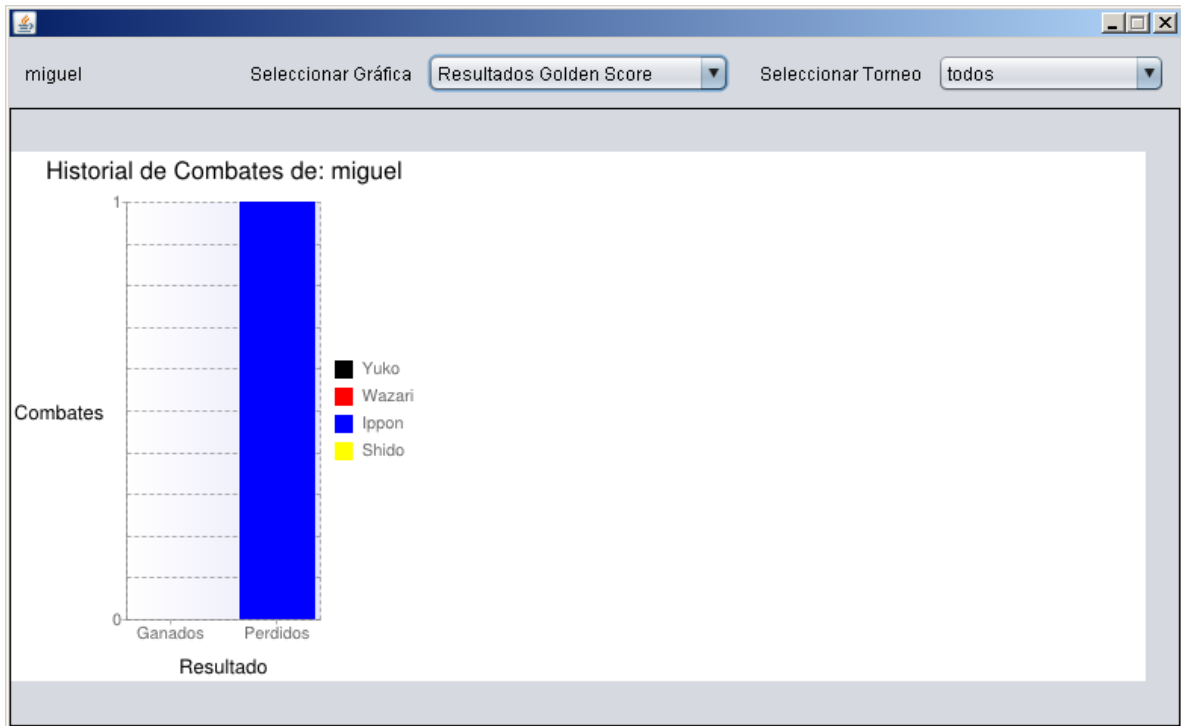


Ilustración 32 GRÁFICA DE COMBATES DE GOLDEN SCORE

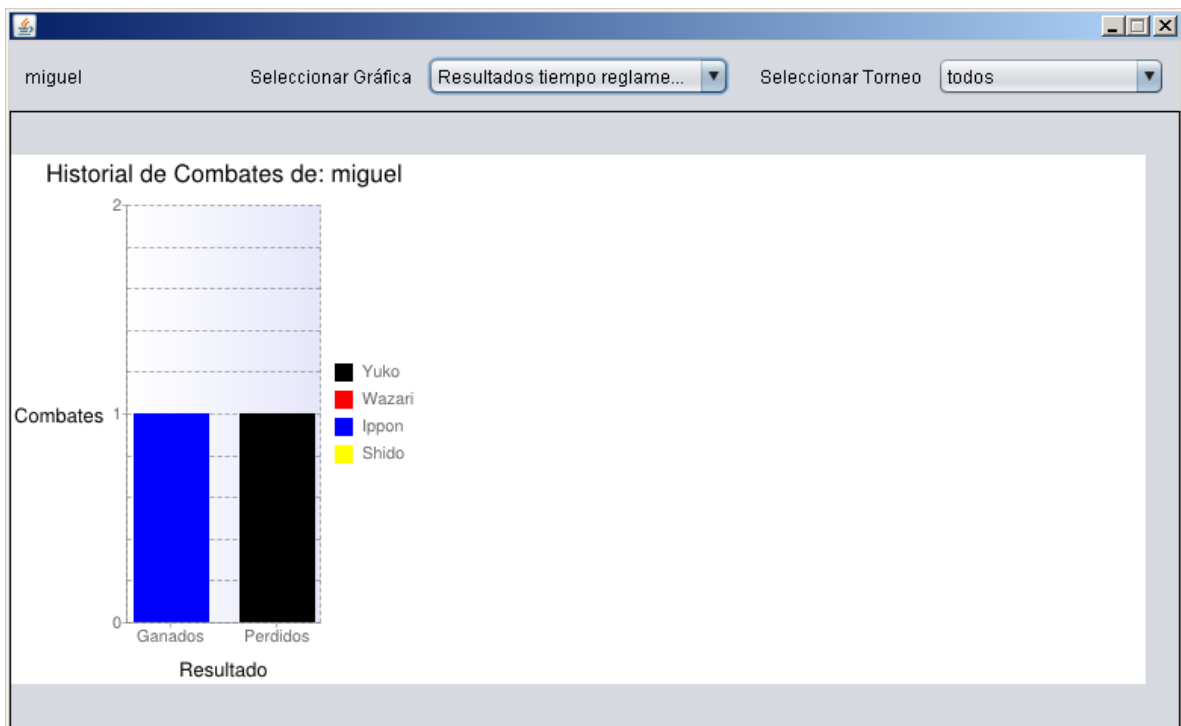


Ilustración 33 GRÁFICA DE COMBATES EN TIEMPO REGLAMENTARIO

Los datos mostrados en las gráficas anteriores pueden estar filtrados por torneo en el cual haya participado el combatiente, si el usuario lo desea.

Por otro lado, de los combates se pueden ver sus datos principales como lugar, ganador, participantes, duración, entre otros. Pero, de este conjunto de datos solo son editables el lugar y el ganador del combate. Adicionalmente, es posible visualizar las gráficas de Frecuencia de ataque, Penalizaciones y Estilo de marcación del combate.

La gráfica de frecuencia de ataque, que se puede ver en la Ilustración 34, permite reflejar la cantidad de ataques que fueron realizados en el tiempo de combate por ambos competidores.

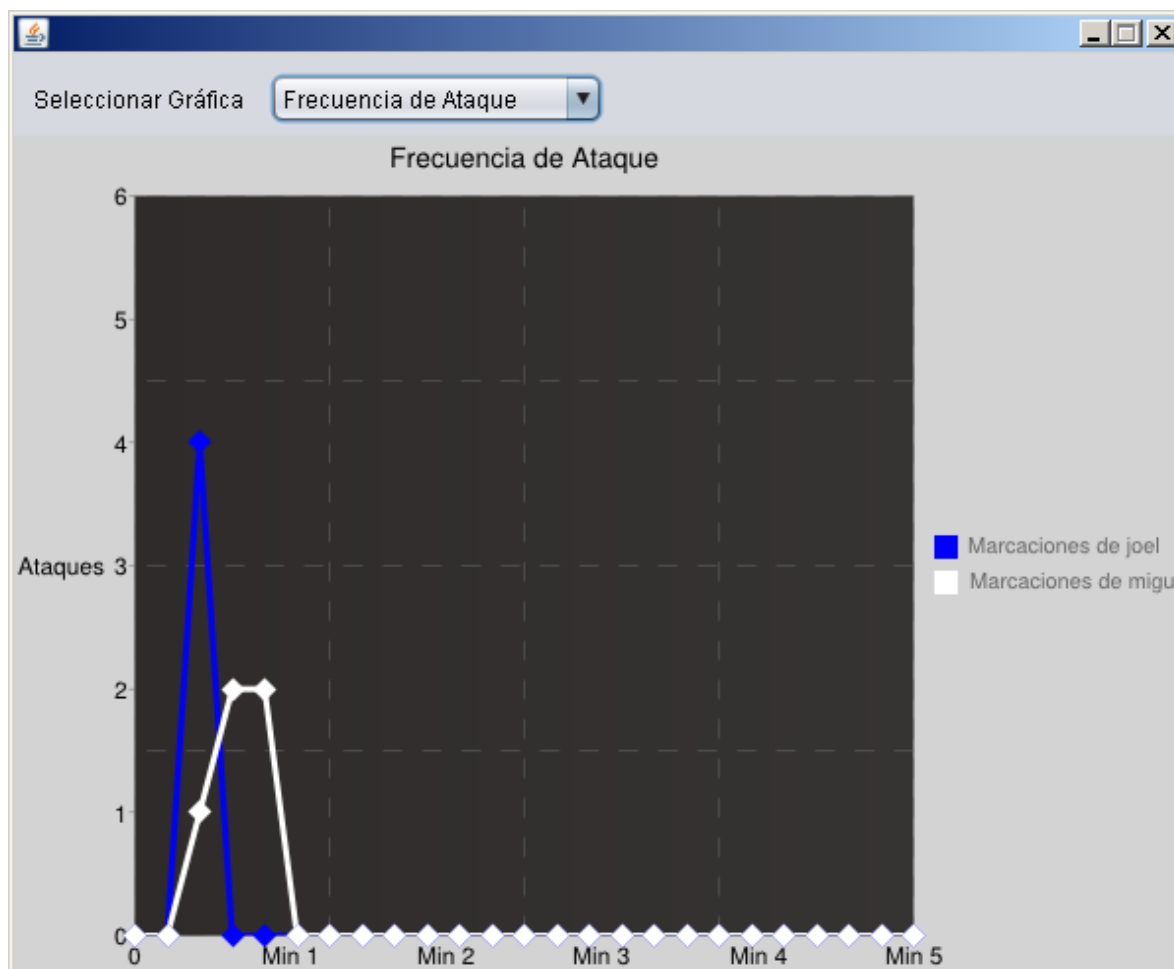


Ilustración 34 GRÁFICA DE FRECUENCIA DE ATAQUE (MARCACIONES) DE UN COMBATE

La gráfica de penalizaciones, detalla las penalizaciones cometidas por ambos combatientes durante la línea de tiempo del combate. En la Ilustración 35 se aprecia un ejemplo de este tipo de gráfica que aporta el sistema.

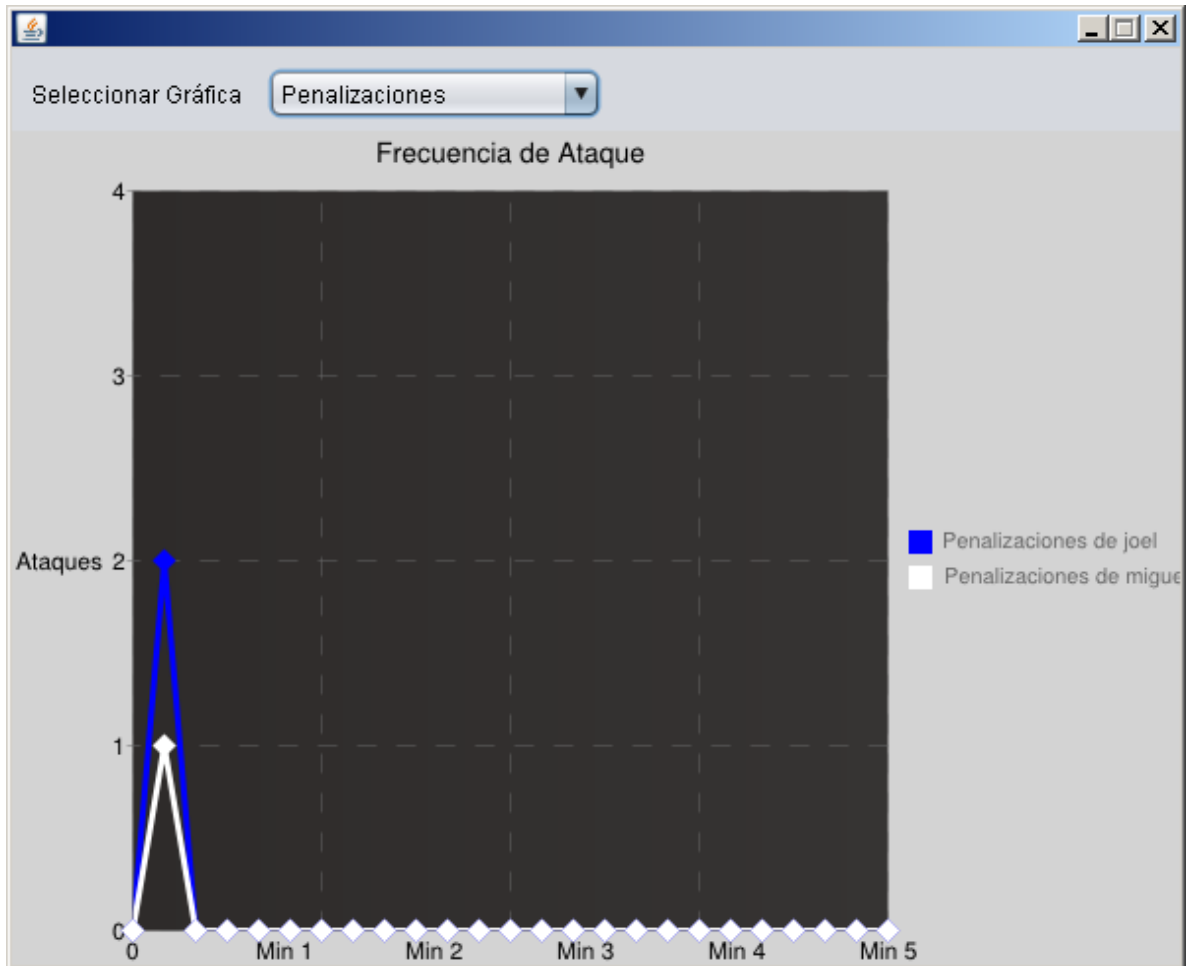


Ilustración 35 GRÁFICA DE FRECUENCIA DE ATAQUE (PENALIZACIONES) DE UN COMBATE

Finalmente, la gráfica estilo de marcación, como muestra la Ilustración 36, indica según los tipos de marcaciones, la cantidad de veces que estas ocurrieron durante el combate.

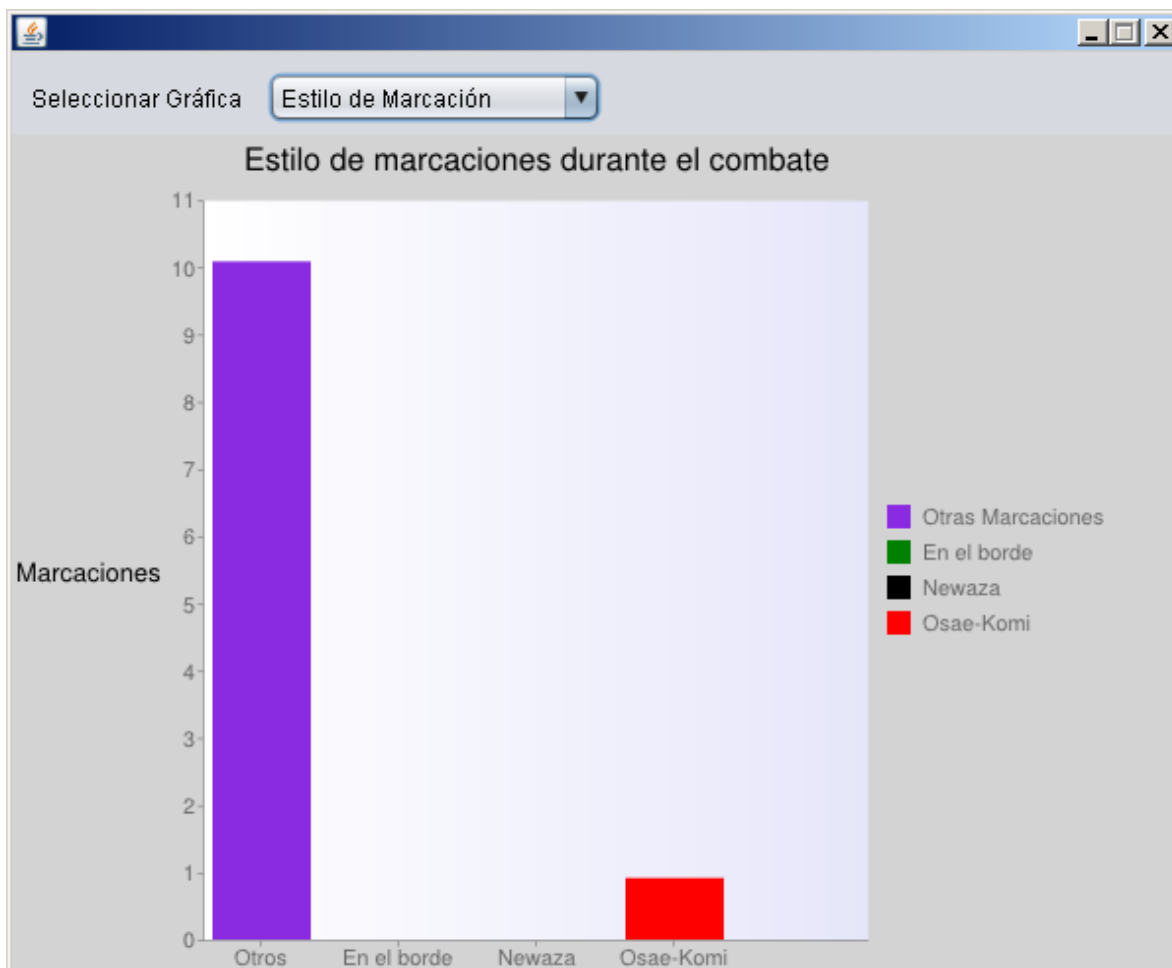


Ilustración 36 GRÁFICA DE ESTILO DE MARCACIÓN

Todas las gráficas correspondientes a los combates pueden ser generadas, de manera individual, como un detalle del combate, o, seleccionando un torneo y graficando todo los combates del mismo.

4.3.2.2 REVISIONES DEL PROTOTIPO INICIAL

Luego de presentar el prototipo inicial al cliente, surgieron modificaciones y requerimientos nuevos, en el caso del combatiente se solicitó agregar la categoría, nacionalidad y entidad venezolana a la cual pertenece, por otro lado, en el caso de los combates se solicitó agregar la categoría a la cual corresponde y un filtro de combates por torneo.

Para la generación de graficas de combatiente, se solicitó que los datos históricos pudieran ser filtrados por torneo. En otro contexto, se solicitó agregar la lógica de torneos y la capacidad de editar y ver su información, tales como, graficas de los combates pertenecientes a un torneo, su descripción y fecha de inicio.

Finalmente, el cliente solicitó modificaciones respecto a la interfaz, para mejorar metáforas escritas relacionadas a funcionalidades del sistema.

4.3.2.3 PROTOTIPO FINAL

Para la solución de *Software* final, se desarrollaron las funcionalidades del prototipo inicial y se agregaron las revisiones y requerimientos solicitados por el cliente. Todo lo anterior enmarcado en aspectos de calidad de *Software* aplicados.

En el aspecto funcional, se agregaron la categoría a los combates y su respectivo filtro, además, como se puede ver en la Ilustración 37, se agregó toda la lógica necesaria para la manipulación de torneos y visualizar las gráficas de los combates en un torneo específico.



Ilustración 37 INTERFAZ PRINCIPAL DEL PROTOTIPO FINAL DE LA APLICACIÓN DE ESCRITORIO

4.3.3 DESARROLLO DE INTERFAZ DE COMUNICACIÓN

El desarrollo de la lógica necesaria para recibir, procesar y almacenar los datos capturados durante un combate de Judo se realizó utilizando lenguaje PHP, los códigos desarrollados están alojados en un servidor Web público, esto permite al dispositivo móvil realizar una petición HTTP en la cual envía los datos utilizando la estructura de JSON por un parámetro POST bien conocido.

En cuanto a las etapas del desarrollo estuvo muy relacionado a la necesidad de agregar transferencia de datos entre la aplicación móvil y la aplicación de escritorio.

4.4 PRUEBAS

Posterior a la fase de desarrollo se procedió a realizar pruebas de aceptación al usuario, para realizar esto se utilizó la herramienta de generación de formularios que brinda Google, con el objetivo de posibilitar un ambiente privado al usuario donde pueda contestar sin interferencia del encuestador.

La prueba estaba dirigida al conjunto de entrenadores que pudieran utilizar la aplicación dentro del Club de Judo de la UCV, de los cuales 3 respondieron la encuesta.

La primera pregunta de la prueba de aceptación fue, ¿Considera usted que la aplicación móvil es fácil de usar?, las respuestas pueden ser observadas en la Ilustración 38.

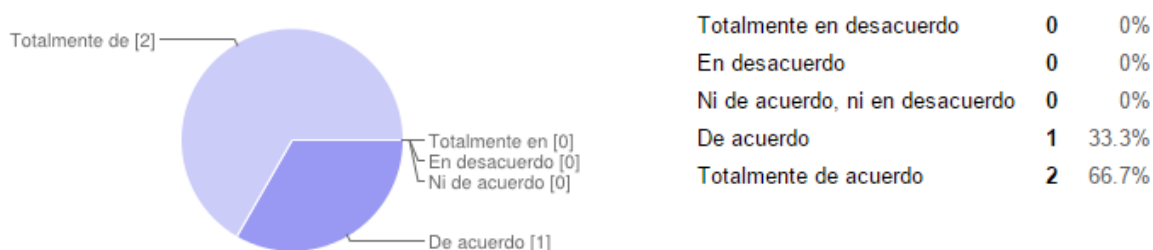


Ilustración 38 RESPUESTAS DE LA PREGUNTA: ¿CONSIDERA USTED QUE LA APLICACIÓN MÓVIL ES FÁCIL DE USAR? EN LA PRUEBA DE ACEPTACIÓN

La segunda pregunta fue, ¿Considera usted que la aplicación móvil permite capturar los eventos ocurridos en el combate y su vídeo asociado?, en la Ilustración 39 puede observarse las respuestas dadas por los entrenadores.

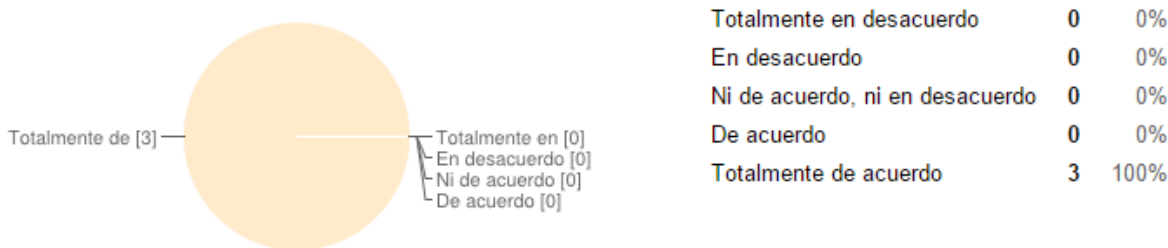


Ilustración 39 RESPUESTAS DE LA PREGUNTA: ¿CONSIDERA USTED QUE LA APLICACIÓN MÓVIL PERMITE CAPTURAR LOS EVENTOS OCURRIDOS EN EL COMBATE Y SU VÍDEO ASOCIADO? EN LA PRUEBA DE ACEPTACIÓN

La tercera pregunta, ¿Considera usted que la aplicación móvil ayuda a agilizar el proceso de recolección de los datos de eventos ocurridos en un combate de Judo?, Las respuestas están reflejadas en la Ilustración 40.



Ilustración 40 RESPUESTAS DE LA PREGUNTA: ¿CONSIDERA USTED QUE LA APLICACIÓN MÓVIL AYUDA A AGILIZAR EL PROCESO DE RECOLECCIÓN DE LOS DATOS DE EVENTOS OCURRIDOS EN UN COMBATE DE JUDO? EN LA PRUEBA DE ACEPTACIÓN

En el caso de la cuarta pregunta, fue la siguiente, ¿Considera usted que los datos que muestra la aplicación de escritorio relacionados a los combates son correctos y se visualizan de una forma adecuada?, en la Ilustración 41 se puede observar las respuestas que brindaron los entrenadores.

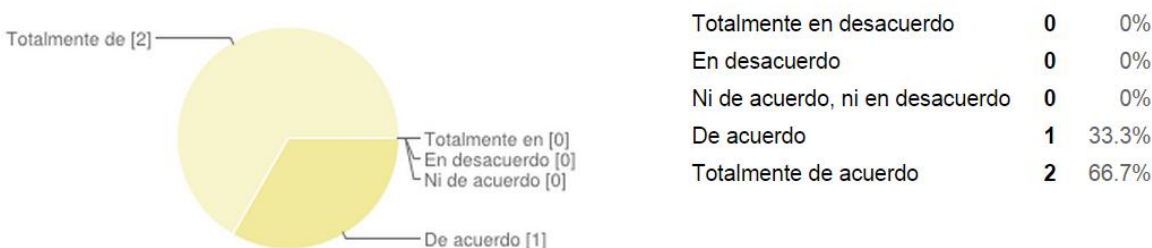


Ilustración 41 RESPUESTAS DE LA PREGUNTA: ¿CONSIDERA USTED QUE LOS DATOS QUE MUESTRA LA APLICACIÓN DE ESCRITORIO RELACIONADOS A LOS COMBATES SON CORRECTOS Y SE VISUALIZAN DE UNA FORMA ADECUADA? EN LA PRUEBA DE ACEPTACIÓN

La quinta pregunta, fue la siguiente, ¿Está usted de acuerdo en que los datos de los combates capturados con la aplicación móvil pueden visualizarse desde la aplicación de escritorio?, las respuestas que dieron los entrenadores se muestran en la Ilustración 42.

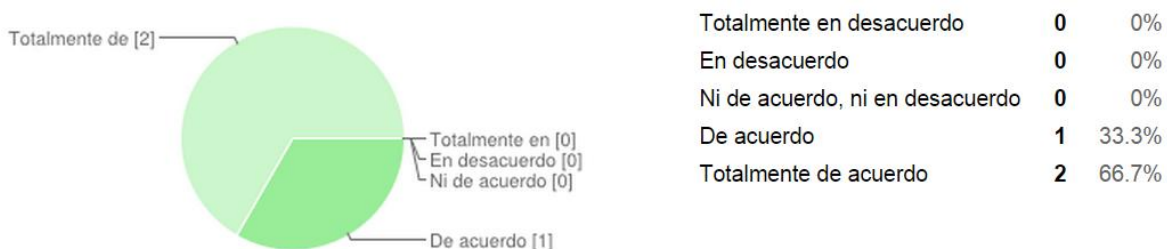


Ilustración 42 RESPUESTAS DE LA PREGUNTA: ¿ESTÁ USTED DE ACUERDO EN QUE LOS DATOS DE LOS COMBATES CAPTURADOS CON LA APLICACIÓN MÓVIL PUEDEN VISUALIZARSE DESDE LA APLICACIÓN DE ESCRITORIO? EN LA PRUEBA DE ACEPTACIÓN

Finalmente, la última pregunta de la prueba fue, ¿Cree usted que la aplicación de escritorio mejora las herramientas que ayudan al análisis de los combates de Judo?, las respuestas se pueden ver en la Ilustración 43.

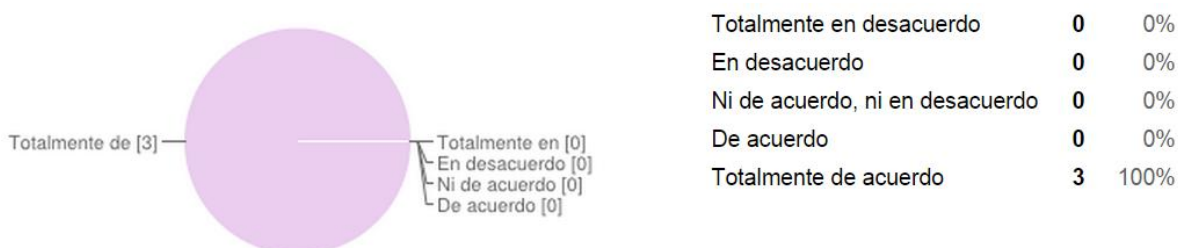


Ilustración 43 RESPUESTAS DE LA PREGUNTA: ¿CREE USTED QUE LA APLICACIÓN DE ESCRITORIO MEJORA LAS HERRAMIENTAS QUE AYUDAN AL ANÁLISIS DE LOS COMBATES DE JUDO? EN LA PRUEBA DE ACEPTACIÓN

Motivado a las respuestas positivas que dieron los entrenadores y las verificaciones en el correcto funcionamiento del sistema, podemos afirmar que la solución de Software desarrollada aprobó las pruebas de aceptación.

4.5 IMPLEMENTACIÓN

Finalizado el desarrollo y las pruebas de aceptación de la aplicación por parte del cliente, se procedió a la entrega de las aplicaciones y la revisión del funcionamiento correcto de la arquitectura de transmisión de los datos. En el caso de la aplicación *Android*, se entregó el archivo que permite la ejecución de la aplicación en formato APK, por otro lado, la aplicación de escritorio se entregó utilizando un empaquetador de archivos bien conocido, cuyo formato de compresión es ZIP, el cual incluye la aplicación en formato JAR y todo el conjunto de bibliotecas necesarias para su funcionamiento. Posteriormente, se

realizó un reconocimiento sobre la arquitectura de transmisión de datos desde la aplicación móvil a la aplicación de escritorio.

CAPÍTULO 5: CONCLUSIONES

Se logró desarrollar la arquitectura de *Software* necesaria para la captura, almacenamiento y visualización de los eventos ocurridos durante un combate de Judo utilizando tecnologías móviles, específicamente *Android*, por su licencia de código abierto y comunidad que permite fácil acceso a la documentación.

En relación al Judo como disciplina deportiva, se estudiaron las reglas y eventos principales que intervienen en el, con la asesoría de entrenadores con años de experiencia en la disciplina. Logrando entender su ejecución y aplicación durante los combates. Además, se comprendió el proceso manual utilizado por los entrenadores para el registro de los eventos ocurridos durante un combate.

Se analizaron las herramientas tecnológicas que permitían desarrollar una solución de software para el problema planteado. Al seleccionar la herramienta acorde se procedió a diseñar la solución basada en un proceso de desarrollo ágil. Seguido a esto, se construyó la solución de software basado en un modelo orientado a prototipos. Finalmente, se realizaron pruebas y la entrega del sistema.

Se obtuvo una significativa mejora en los procesos de registro, almacenamiento y análisis de los eventos ocurridos en un combate de Judo, todo esto gracias a la aplicación de la solución tecnológica resultado del presente trabajo especial de grado.

La aplicación de herramientas tecnológicas permite mejorar los procesos de registro en competencias deportivas, esto depende de un correcto estudio de la disciplina y sus procesos.

5.1 CONTRIBUCIÓN

Este trabajo de investigación contribuye con el deporte Venezolano e impulsa la computación ubicua en el área de deporte lo cual tiene un gran impacto social, adicionalmente, potencia la habilidad de captura, almacenamiento y visualización de los datos generados por los combates de Judo.

Por otro lado, la solución tecnológica producto del presente trabajo de investigación innova en el área de *Software* orientado al deporte en Venezuela, ya que, actualmente no hay registros de aplicaciones orientadas a la captura, almacenamiento y visualización de datos generados durante un combate de Judo.

5.2 RECOMENDACIONES

Para utilizar la herramienta desarrollada con éxito es recomendable tener una conexión de datos disponible, bien sea otorgada por un operador de telecomunicaciones o

por WiFi. Por otro lado, es recomendable sincronizar el dispositivo luego de cada combate para liberar la memoria del mismo, además, en caso de necesitar ver los datos siguiente a un combate, podemos recomendar tener un computador portátil con capacidad de utilizar la aplicación y que también cuente con conexión de Internet.

Para mejorar el sistema, puede ser recomendable buscar un apoyo financiero para contratar un servicio de almacenamiento en la nube que tenga mayor capacidad, velocidad y respaldo garantizado.

Para el correcto funcionamiento de la solución de *Software* es necesario utilizar un dispositivo móvil con una pantalla de al menos siete (7) pulgadas, que disponga de cámara posterior, ideal de al menos 5 megapíxeles, con sistema operativo *Android* cuya versión sea 4.0 *Ice Cream Sandwich* o superior. En el caso de la aplicación de escritorio, es preciso tener la última versión del sistema operativo, bien sea *Windows* o *Linux*, que tenga el JRE de Java actualizado en su última versión, adicionalmente, debe tener los requerimientos básicos de hardware para el correcto funcionamiento del sistema operativo.

5.3 LÍMITES

Debido a la complejidad de la arquitectura y de las herramientas desarrolladas este trabajo de investigación se vio limitado en el desarrollo de algunos requerimientos, tales como, optimización de almacenamiento de videos. Por otro lado, por razones técnicas de capacidad de memoria volátil y límites teóricos en cuanto al espacio de almacenamiento del dispositivo, la grabación de la pelea está limitada a 15 minutos.

Motivado a las bibliotecas que utiliza la aplicación para reproducir el material multimedia capturado es necesario que el computador donde este ejecutándose la aplicación de escritorio tenga instalado el programa *VLC media player*, por otro lado, también es indispensable que el computador tenga una conexión a Internet para poder acceder al servidor de datos.

Producto de las dependencias de algunas funcionalidades de la biblioteca del SDK de *Android* es necesario que la versión del sistema operativo del dispositivo sea al menos 4.0, esto para poder ejecutar la aplicación en dicho terminal móvil correctamente.

La interconexión de las aplicaciones utiliza servicios de Google, por lo tanto es necesario que exista una cuenta de *Gmail* dedicada a almacenar el contenido multimedia y debe estar asociada en ambas aplicaciones. Adicionalmente, el dispositivo móvil debe tener instalado la aplicación de Servicios de Google, ya que en dicha aplicación se encuentran bibliotecas que son utilizadas para la sincronización de los datos.

Además, el esquema de sincronización que fue creado solamente permite el correcto funcionamiento de las aplicaciones cuando un solo dispositivo móvil es usado simultáneamente.

5.4 TRABAJOS FUTUROS

Este trabajo de investigación al ser aplicado por la selección nacional de Judo podría generar gran cantidad de información, sería interesante aplicar técnicas para recolectar más información de dichos datos, por ejemplo implementar minería de datos y así obtener patrones de ataque, movimientos comunes, vías efectivas de entrenamiento, entre otros.

Adicionalmente, puede ser productivo e interesante recolectar más información del combate, por ejemplo datos fisiológicos de los combatientes, utilizando dispositivos inalámbricos que se conecten a la aplicación que es ejecutada en la Tablet y que luego sean sincronizados al servidor para su posterior análisis e interpretación en el computador.

En cuanto a la sincronización, una posible propuesta muy interesante sería, generar una arquitectura de servicios web bien definida, que además permita la interacción de N dispositivos móviles con distintas cuentas de *Gmail* asociadas.

En relación a la reproducción de videos, es posible desarrollar una biblioteca para visualizar el material multimedia capturado por el dispositivo móvil, esto con el objetivo de no utilizar las bibliotecas de VLC, lo cual haría que la portabilidad de la aplicación se viese incrementada.

Con el objetivo de medir el impacto de la aplicación en el rendimiento de los Judocas, sería productivo un trabajo de investigación que evalúe dicho rendimiento antes y después de la utilización del sistema generado en el presente trabajo.

Una funcionalidad muy útil que podría agregarse al sistema es la capacidad de capturar un combate sin grabarlo con la cámara del dispositivo móvil y luego poder agregar el video desde la aplicación de escritorio.

Motivado al éxito de la investigación en la disciplina de Judo, podría aplicarse este método para desarrollar soluciones tecnológicas para otras disciplinas, estudiando cada caso de forma particular, adicionalmente, puede realizarse un estudio de aplicabilidad con el objetivo de construir una aplicación genérica para un conjunto de disciplinas que sean de estilo similar.

Para enriquecer el aplicativo, sería útil traducir las aplicaciones y manuales generados de esta investigación, al menos a los lenguajes principales usados por entrenadores y atletas de Judo.

REFERENCIAS

- [1] IFJ, «International Judo Federation,» 2014. [En línea]. Available: http://www.intjudo.eu/Judo_Corner. [Último acceso: 2014].
- [2] S. L, «EFDeportes.com,» 16 Febrero 2012. [En línea].
- [3] C. T. Jorge, «Inteligencia de redes de comunicaciones, minería de datos en el deporte profesional,» Madrid, 2012.
- [4] I. Chavel, J. Dopico, E. Iglesias y A. Rivas, «Propuesta de Metodología para el estudio y análisis de la estructura temporal del enfrentamiento en Judo,» Coruña.
- [5] International Data Corporation, «IDC: Smartphone OS Market Share 2014, 2013, 2012, and 2011,» 2014. [En línea]. Available: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>.
- [6] Android, «Android, the world's most popular mobile platform,» 2014. [En línea]. Available: <http://developer.android.com/about/index.html>.
- [7] Android, «developer.android.com,» 2014 septiembre. [En línea]. Available: <http://developer.android.com/tools/sdk/eclipse-adt.html>.
- [8] Android, «Sharing Simple Data | Android Developers,» [En línea]. Available: <https://developer.android.com/training/sharing/index.html>. [Último acceso: 02 2015].
- [9] Google, «Introduction to the Google Drive API,» 02 2015. [En línea]. Available: <https://developers.google.com/drive/android/intro>. [Último acceso: 02 2015].
- [10] Android, «Android, Mecanismos de comunicación y manipulación de la cámara del dispositivo.,» [En línea]. Available: <http://developer.android.com/guide/topics/media/camera.html>.
- [11] Android, «Android, especificación de CountdownTimer,» [En línea]. Available: <http://developer.android.com/reference/android/os/CountDownTimer.html>.
- [12] J. Gosling, B. Joy, G. Steele, G. Bracha y A. Buckley, «The Java® Language Specification Java SE 8 Edition,» Oracle America, Redwood, California, U.S.A., 2014.
- [13] Oracle Inc., «Learn about Java Technology,» 14 02 2015. [En línea]. Available: <https://www.java.com/en/about/>. [Último acceso: 02 2015].
- [14] the PHP Documentation Group, «PHP Manual,» 02 2015. [En línea]. Available: <http://php.net/manual/en/>. [Último acceso: 02 2015].
- [15] NetBeans, «NetBeans Docs and Support,» 2013. [En línea]. Available: <https://netbeans.org/kb/index.html>.

- [16] Rivieres y J. des, «eclipse.org,» 2006. [En línea]. Available: <https://www.eclipse.org/articles/Whitepaper-Platform-3.1/eclipse-platform-whitepaper.html>.
- [17] Android, «Android Studio Overview,» Febrero 2015. [En línea]. Available: <http://developer.android.com/tools/studio/>.
- [18] ECMA internacional, «ECMA-404 The JSON Data Interchange Format,» 10 2013. [En línea]. Available: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>. [Último acceso: 02 2015].
- [19] Google, « A Java library to convert JSON to Java objects and vice-versa,» [En línea]. Available: <https://code.google.com/p/google-gson/>. [Último acceso: 02 2015].
- [20] J. C, «charts4j - Frequently Asked Questions,» 6 5 2011. [En línea]. Available: <https://code.google.com/p/charts4j/wiki/FAQ>. [Último acceso: 10 2 2015].
- [21] J. Chastang, «charts4j API documentation,» 19 5 2009. [En línea]. [Último acceso: 11 02 2015].
- [22] S. Narayan, «Sugar ORM - Insanely easy way to work with Android database.,» 2014. [En línea]. Available: <http://satyan.github.io/sugar/>. [Último acceso: 12 02 2015].
- [23] The Apache Software Foundation, «About the Apache HTTP Server Project,» 2015. [En línea]. Available: <http://httpd.apache.org/>. [Último acceso: 11 02 2015].
- [24] D. Eastlake y T. Hansen, «RFC 6234 - US Secure Hash Algorithms,» 5 2011. [En línea]. Available: <http://tools.ietf.org/html/rfc6234>. [Último acceso: 11 02 2015].
- [25] S. Annies, «mp4parser,» 10 10 2013. [En línea]. Available: <https://code.google.com/p/mp4parser/>. [Último acceso: 10 03 2015].
- [26] Google, «MediaRecorder | Android Developers,» 20 2 2015. [En línea]. Available: <http://developer.android.com/reference/android/media/MediaRecorder.html>. [Último acceso: 10 3 2015].
- [27] SQLite, «About SQLite,» 02 2015. [En línea]. Available: <http://www.sqlite.org/about.html>. [Último acceso: 02 2015].
- [28] MySQL, «MySQL 5.7 Reference Manual,» 02 2015. [En línea]. Available: <http://dev.mysql.com/doc/refman/5.7/en/>. [Último acceso: 02 2015].
- [29] Google, «Drive,» 2015. [En línea]. Available: <https://support.google.com/drive/>. [Último acceso: 12 02 2015].
- [30] IJF, «International Judo Federation. Refereeing Rules 2014-2016.,» 2014. [En línea]. Available:

http://www.intjudo.eu/upload/2014_06/03/140180706544806028/ijf_ref_rules_2014_16_eng_final_version_1.05.pdf.

- [31] I. Sommerville, Ingeniería del Software. Séptima Edición, Madrid: Pearson Educación S.A., 2005.
- [32] O. Rosales, «MODELOS DE PROCESOS DE LA INGENIERIA DE SOFTWARE,» 27 5 2012. [En línea]. Available: <http://www.scribd.com/doc/94934934/MODELOS-DE-PROCESOS-DE-INGENIERIA-DE-SOFTWARE>. [Último acceso: 10 03 2015].
- [33] R. S. Pressman y B. R. Maxim, Software Engineering: a practitioner's approach, New York: McGraw-Hill Education, 2015.
- [34] Android, «Documentación oficial de SDK Android,» 2014. [En línea]. Available: <https://developer.android.com/tools/help/index.html>. [Último acceso: 2014].

ANEXOS

	1er minuto			2do minuto			3er minuto			4to minuto			Golden Score	TOTALES
	Y	W	I	Y	W	I	Y	W	I	Y	W	I		
1er Agarre														
Ataque Brazo														
Ataque cadera														
Ataque Pierna														
Pasividad														
Fuera del Área														
Ataque Falso														
Descalificación														
Ne waza														
Mate														
1er Agarre														
Ataque Brazo														
Ataque cadera														
Ataque pierna														
Pasividad														
Fuera del Área														
Ataque Falso														
Descalificación														
Ne waza														

SJFT	I= FC final - FC1 Nro Proyecciones	Fc Final	Fc 1min	proyecciones		17-abr								
		188	169	28	12,75	131	ANDREA GOMEZ							
		184	158	25	13,68	117	NAIBETH							
		191	170	25	14,44	150	YORMARI							
		177	158	27	12,40741	139	MORILLO							
		186	152	24	14,08333	112	JULIENI							
		187	170	24	14,875	134	ANDRIMAR							
		189	162	27	13	112	GISELA							
		188	168	23	15,47826	124	MILAGRO	121	187	164	28	12,53571		JULIO= 2014
		181	164	31	11,12903	91	CORTESIA	88	187	168	29	12,24138		
		188	152	23	14,78261	89	DIANA	108	186	164	26	13,46154		
		173	142	29	10,86207	114	ANRIQUEL	97	177	145	28	11,5		
		190	172	26	13,92308	108	WISNEIBIS	123	189	167	29	12,27586		
							ELVISMAF	99	174	149	26	12,42308		

Table 5. Classificatory norms for total number of throws, heart rate (after and 1-min after) and index in the Special Judo Fitness Test (n=141) [17]

Classification	Variables			
	Total of throws	HR after (bpm)	HR 1min after (bpm)	Index
Excellent	≥29	≤173	≤143	≤11.73
Good	27-28	174-184	144-161	11.74-13.03
Average	26	185-187	162-165	13.04-13.94
Poor	25	188-195	166-174	13.95-14.84
Very Poor	≤24	≥196	≥175	≥14.85

HR – heart rate

HOJA DE CÁLCULO UTILIZADA ACTUALMENTE POR EL ENTRENADOR DE LA SELECCIÓN DE JUDO

Columna	Tipo	Nulo	Pre de terminado	Comentarios	MIME
id	int(11)	No		Clave primaria de la tabla	
descripcion	varchar(255)	No		nombre de la categoría	
fecha_creacion	timestamp	No	CURRENT_TIMESTAMP	fecha de creación de cada fila	
fecha_modificacion	timestamp	No	0000-00-00 00:00:00	fecha de modificación de cada fila	
id_sync	int(11)	No		indica el id que la fila tiene en el dispositivo movil	
sync	tinyint(1)	No		indica si la fila debe ser sincronizada o no	

DICCIONARIO DE DATOS DE LA TABLA CATEGORÍAS EN LA BASE DE DATOS MYSQL

Columna	Tipo	Nulo	Pre de terminado	Comentarios	MIME
id	int(11)	No		clave primaria de la tabla	
CompetidorA	int(11)	No		clave foránea de competidor	
CompetidorB	int(11)	No		clave foránea de competidor	
Ganador	int(11)	No		clave foránea de competidor	
Lugar	text	No		indica el lugar del combate	
Duracion	int(10)	No		indica la duración del combate	
Eventos	int(10)	No		indica el número de eventos del combate	
Fecha	datetime	No		indica la fecha del combate	
GoldenScore	tinyint(1)	No	0	indica si el combate fue a "Golden Score"	
idCategoria	int(11)	No		clave foránea de categoría	
idTorneo	int(11)	No		clave foránea de torneo	
id_sync	int(11)	No		indica el id de la fila en el dispositivo movil	
sync	tinyint(1)	No		indica si la fila debe ser sincronizada o no	
fecha_creacion	timestamp	No	CURRENT_TIMESTAMP	fecha de creación de la fila	
fecha_modificacion	timestamp	No	0000-00-00 00:00:00	fecha de modificación de la fila	

DICCIONARIO DE DATOS DE LA TABLA COMBATES EN LA BASE DE DATOS MYSQL

Columna	Tipo	Nulo	Pre de terminado	Comentarios	MIME
id	int(11)	No		clave primaria de la fila	
nombre	text	No		indica el nombre del competidor	
sexo	tinyint(1)	No		indica el sexo del competidor	
peso	int(3)	Sí	NULL	indica el peso del competidor	
edad	int(3)	Sí	NULL	indica la edad del competidor	
estatura	decimal(3,0)	Sí	NULL	indica la estatura del competidor	
tipoCombate	int(2)	No	0	indica el estilo de combate del competidor	
seleccion	tinyint(1)	Sí	NULL	indica si el competidor es de la seleccion venezolana o no	
idCategoria	int(11)	No		clave foránea de categoria	
nacionalidad	int(11)	No		clave foranea de nacionalidad	
entidad	int(11)	No		clave foranea de entidad	
fecha_creacion	timestamp	No	CURRENT_TIMESTAMP	fecha de creación de la fila	
fecha_modificacion	timestamp	No	0000-00-00 00:00:00	fecha de modificación de la fila	
id_sync	int(11)	No	0	indica el id de la fila en el dispositivo movil	
sync	tinyint(1)	No		indica si la fila debe ser sincronizada o no	

DICCIONARIO DE DATOS DE LA TABLA COMPETIDOR EN LA BASE DE DATOS MYSQL

Columna	Tipo	Nulo	Pre de terminado	Comentarios	MIME
id	int(11)	No		clave primaria de la fila	
nombre	varchar(60)	No		nombre del estado	

DICCIONARIO DE DATOS DE LA TABLA ESTADO EN LA BASE DE DATOS MYSQL

Columna	Tipo	Nulo	Pre de terminado	Comentarios	MIME
id	int(11)	No		clave primaria de la fila	
nombre	varchar(100)	No		nombre del evento	

DICCIONARIO DE DATOS DE LA TABLA MARCACIÓN EN LA BASE DE DATOS MYSQL

Columna	Tipo	Nulo	Pre de terminado	Comentarios	MIME
id	int(11)	No		clave primaria de la fila	
iso	char(2)	No		nombre del pais con formato	
name	varchar(80)	No		nombre del pais con formato	
nicename	varchar(80)	No		nombre del pais con formato	
iso3	char(3)	Si	NULL	nombre del pais con formato	
numcode	smallint(6)	Si	NULL	codigo numerico del pais	
phonecode	int(5)	No		codigo de telefono del pais	

DICCIONARIO DE DATOS DE LA TABLA NACIONALIDAD EN LA BASE DE DATOS MYSQL

Columna	Tipo	Nulo	Pre de terminado	Comentarios	MIME
id	int(11)	No		clave primaria de la fila	
ruta	varchar(255)	Si	NULL	indica la ruta del video	
nombre	text	Si	NULL	indica el nombre del archivo de video	
subido	tinyint(1)	Si	NULL	indica si la fila debe ser sincronizada	
idCombate	int(11)	Si	NULL	clave foránea de combate	

DICCIONARIO DE DATOS DE LA TABLA VIDEO EN LA BASE DE DATOS MYSQL

Columna	Tipo	Nulo	Pre de terminado	Comentarios	MIME
id	int(11)	No		clave primaria de la fila	
descripcion	varchar(255)	No		indica el nombre del torneo	
fecha_creacion	timestamp	No	CURRENT_TIMESTAMP	fecha de creación de la fila	
fecha_modificacion	timestamp	No	0000-00-00 00:00:00	fecha de modificación de la fila	
id_sync	int(11)	No		indica el id de la fila en el dispositivo movil	
sync	tinyint(1)	No		indica si la fila debe ser sincronizada o no	

DICCIONARIO DE DATOS DE LA TABLA TORNEO EN LA BASE DE DATOS MYSQL

Columna	Tipo	Nulo	Pre de terminado	Comentarios	MIME
id	int(11)	No		clave primaria de la fila	
idCombate	int(11)	Sí	NULL	clave foránea de combate	
receptor	int(11)	Sí	NULL	clave foránea de competidor	
ejecutor	int(11)	Sí	NULL	clave foránea de competidor	
descripcion	int(11)	Sí	NULL	indica descripción del evento	
instante	bigint(20)	Sí	NULL	indica el instante de tiempo en que ocurrió el evento	
id_sync	int(11)	No		indica el id de la fila en el dispositivo movil	
sync	tinyint(1)	No		indica si la fila debe ser sincronizada o no	
comentario	text	No		indica el comentario en el evento	
idEventoRelacionado	int(11)	No	0	clave foránea de evento	
isOsaekomi	tinyint(1)	No	0	indica si el evento fue osaekomi	
isBorder	tinyint(1)	No		indica si el evento fue en el borde	
isNewaza	tinyint(1)	No		indica si el evento fue en newaza	
fecha_creacion	timestamp	No	CURRENT_TIMESTAMP	fecha de creación de la fila	
fecha_modificacion	timestamp	No	0000-00-00 00:00:00	fecha de modificación de la fila	

DICCIONARIO DE DATOS DE LA TABLA EVENTOS EN LA BASE DE DATOS MYSQL

Columna	Tipo	Nulo	Pre de te rminado	Comentarios	MIME
id	int(11)	No		Clave primaria de la tabla	
descripcion	varchar(255)	No		nombre de la categoría	
sync	tinyint(1)	No		indica si la fila debe ser sincronizada o no	

DICCIONARIO DE DATOS DE LA TABLA CATEGORÍA EN LA BASE DE DATOS SQLITE

Columna	Tipo	Nulo	Pre de te rminado	Comentarios	MIME
id	int(11)	No		clave primaria de la tabla	
CompetidorA	int(11)	No		clave foránea de competidor	
CompetidorB	int(11)	No		clave foránea de competidor	
Ganador	int(11)	No		clave foránea de competidor	
Lugar	text	No		indica el lugar del combate	
Duracion	int(10)	No		indica la duración del combate	
Eventos	int(10)	No		indica el número de eventos del combate	
Fecha	datetime	No		indica la fecha del combate	
idCategoría	int(11)	No		clave foránea de categoría	
idTorneo	int(11)	No		clave foránea de torneo	
sync	tinyint(1)	No		indica si la fila debe ser sincronizada o no	

DICCIONARIO DE DATOS DE LA TABLA COMBATES EN LA BASE DE DATOS SQLITE

Columna	Tipo	Nulo	Pre de te rminado	Comentarios	MIME
id	int(11)	No		clave primaria de la fila	
nombre	text	No		indica el nombre del competidor	
sexo	tinyint(1)	No		indica el sexo del competidor	
tipoCombate	int(2)	No	0	indica el estilo de combate del competidor	
idCategoría	int(11)	No		clave foránea de categoría	
nacionalidad	int(11)	No		clave foránea de nacionalidad	
entidad	int(11)	No		clave foránea de entidad	
sync	tinyint(1)	No		indica si la fila debe ser sincronizada o no	

DICCIONARIO DE DATOS DE LA TABLA COMPETIDOR EN LA BASE DE DATOS SQLITE

Columna	Tipo	Nulo	Pre de terminado	Comentarios	MIME
id	int(11)	No		clave primaria de la fila	
nombre	varchar(60)	No		nombre del estado	

DICCIONARIO DE DATOS DE LA TABLA ESTADO EN LA BASE DE DATOS SQLITE

Columna	Tipo	Nulo	Pre de terminado	Comentarios	MIME
id	int(11)	No		clave primaria de la fila	
idCombate	int(11)	Sí	NULL	clave foránea de combate	
receptor	int(11)	Sí	NULL	clave foránea de competidor	
ejecutor	int(11)	Sí	NULL	clave foránea de competidor	
descripcion	int(11)	Sí	NULL	indica descripción del evento	
instante	bigint(20)	Sí	NULL	indica el instante de tiempo en que ocurrió el evento	
sync	tinyint(1)	No		indica si la fila debe ser sincronizada o no	
idEventoRelacionado	int(11)	No	0	clave foránea de evento	

DICCIONARIO DE DATOS DE LA TABLA EVENTO EN LA BASE DE DATOS SQLITE

Columna	Tipo	Nulo	Pre de terminado	Comentarios	MIME
id	int(11)	No		clave primaria de la fila	
nombre	varchar(100)	No		nombre del evento	

DICCIONARIO DE DATOS DE LA TABLA MARCACIÓN EN LA BASE DE DATOS SQLITE

Columna	Tipo	Nulo	Pre de terminado	Comentarios	MIME
id	int(11)	No		clave primaria de la fila	
iso	char(2)	No		nombre del país con formato	
name	varchar(80)	No		nombre del país con formato	
nicename	varchar(80)	No		nombre del país con formato	
iso3	char(3)	Sí	NULL	nombre del país con formato	
numcode	smallint(6)	Sí	NULL	código numerico del país	
phonecode	int(5)	No		código de teléfono del país	

DICCIONARIO DE DATOS DE LA TABLA NACIONALIDAD EN LA BASE DE DATOS SQLITE

Columna	Tipo	Nulo	Pre de terminado	Comentarios	MIME
id	int(11)	No		clave primaria de la fila	
descripcion	varchar(255)	No		indica el nombre del torneo	
sync	tinyint(1)	No		indica si la fila debe ser sincronizada o no	

DICCIONARIO DE DATOS DE LA TABLA TORNEO EN LA BASE DE DATOS SQLITE

Columna	Tipo	Nulo	Pre de terminado	Comentarios	MIME
id	int(11)	No		clave primaria de la fila	
ruta	varchar(255)	Sí	<i>NULL</i>	indica la ruta del video	
nombre	text	Sí	<i>NULL</i>	indica el nombre del archivo de video	
subido	tinyint(1)	Sí	<i>NULL</i>	indica si la fila debe ser sincronizada	
idCombate	int(11)	Sí	<i>NULL</i>	clave foránea de combate	

DICCIONARIO DE DATOS DE LA TABLA VIDEO EN LA BASE DE DATOS SQLITE