



PROYECTO DE GRADO

Presentado ante la ilustre UNIVERSIDAD DE LOS ANDES como requisito parcial para obtener el Título de INGENIERO DE SISTEMAS

www.bdigital.ula.ve

SOFTWARE DE ANALISIS E IMPLEMENTACION TACTICA PARA ENTRENADORES DEPORTIVOS

POR

BR. DARWIN DARIO GONZALEZ BECERRA

TUTOR: DR. GERARD PAEZ MONZON

Mérida, Noviembre de 2022

C.C. Reconocimiento

Resumen

El presente trabajo tiene como objetivo el desarrollo de un software de análisis e implementación táctica que pueda ser utilizado por entrenadores deportivos para crear, simular y guardar sistemas de juego, de entrenamiento y tácticos. El software se desarrolló siguiendo la metodología ágil 'Extreme Programming', así como sus fases y las herramientas que componen a esta metodología. El software está concebido como una herramienta para ser integrada en plataformas digitales de clubes deportivos, por tal motivo en la implementación se utilizaron las tecnologías HTML, CSS, JavaScript que son de simple integración con otros proyectos. Al seguir las fases de la metodología ya mencionada se obtuvo un software con tres funcionalidades principales las cuales permiten crear sobre un ambiente de trabajo una secuencia de movimientos deseada, guardar la secuencia y poder recrearla, así como ser capaz de editarla. Estas funcionalidades demostraron un alto nivel de desarrollo y pueden compararse a funcionalidades de software profesionales de este ámbito

Palabras clave: Software, táctica, metodología ágil, Extreme programming

Índice general

Agradecimientos	III
Resumen	IV
Índice de Figuras	VII
Índice de Cuadro	IX
1. Introducción	1
1.1. Antecedentes	1
1.2. Planteamiento del Problema	3
1.3. Objetivos	5
1.3.1. Objetivo General	5
1.3.2. Objetivos Específicos	5
1.4. Justificación	6
1.5. Propósito	7
2. Marco Teórico	8
2.1. Bases Teóricas	8
2.1.1. Entrenamiento Deportivo	8
2.1.1.1. Sistema de Entrenamiento	9
2.1.2. Método de Entrenamiento	10
2.1.3. Táctica Deportiva	10
2.1.4. Sistemas de Juego	11
2.1.5. Metodología Desarrollo de Software	11
2.1.5.1. Metodologías Tradicionales	12

2.1.5.2. Metodologías Agiles	12
2.1.5.3. Metodología Extreme Programming (XP)	12
2.1.6. JavaScript	18
2.1.7. HTML	19
2.1.8. CSS	19
2.1.9. Framework	19
2.1.9.1. Konva	20
2.1.10. Wireframe	20
2.1.11. Bases de Datos	21
2.1.11.1. Bases de Datos Relacionales	21
2.1.11.2. Bases de Datos No Relacionales	21
2.1.11.2.1. MongoDB	22
3. Marco Metodológico	23
3.1. Diseño de Investigación	23
3.2. Tipo de Investigación	24
3.3. Nivel de Investigación	24
3.4 Metodología a Utilizar	25
4. Desarrollo del Proyecto	26
4.1 Planificación	26
4.1.1 Historias de Usuario	28
4.1.2 Tareas de Ingeniería	31
4.2 Diseño	37
4.3 Codificación	41
4.5 Pruebas	47
4.5.1 Pruebas de Aceptación	47
4.5.1 Comparativa entre Softwares	56
5. Conclusiones y Recomendaciones	60
Bibliografía	62

Índice de figuras

4.1. Vista Software profesional <i>TheCoachPlatform</i>	27
4.2. Vista Software profesional <i>TacticalPad</i>	27
4.3. Boceto Funcionalidad 1	37
4.4. Boceto Funcionalidad 2	38
4.5. Boceto Funcionalidad 3	39
4.6. Boceto Funcionalidad 4	40
4.7. Primera parte Implementación Funcionalidad 1	42
4.8. Segunda parte Implementación Funcionalidad 1	42
4.9. Implementación Funcionalidad 2	43
4.10. Primera parte Implementación Funcionalidad 3	44
4.11. Segunda parte Implementación Funcionalidad 3	45
4.12. Implementación Funcionalidad 4	46
4.13. Modelo de datos de una Jugada	47
4.14. Primera parte prueba de aceptación 1	49
4.15. Segunda parte prueba de aceptación 1	49
4.16. Tercera parte prueba de aceptación 1	50
4.17. Cuarta parte Prueba de aceptación 1	50
4.18. Primera parte Prueba de aceptación 2	51
4.19. Segunda parte Prueba de aceptación 2	52
4.20. Primera parte Prueba de aceptación 3	53
4.21. Segunda parte Prueba de aceptación 3	54
4.22. Tercera parte Prueba de aceptación 3	54
4.23. Prueba de aceptación 4	55

4.24. Ejercicio de enteramiento utilizando el software desarrollado	56
4.25. Ejercicio de enteramiento utilizando el software <i>TactitalPad</i>	57
4.26. Ejercicio de enteramiento utilizando el software <i>TheCoachPlatform</i>	57
4.27. Ejercicio táctico utilizando el software desarrollado	58
4.28. Ejercicio táctico utilizando el software <i>TactitalPad</i>	59
4.29. Ejercicio táctico utilizando el software <i>TheCoachPlatform</i>	59

www.bdigital.ula.ve

Índice de Cuadros

2.1. Plantilla para Historias de Usuario	16
2.2. Plantilla para Tareas de Ingeniera	17
2.3. Plantilla para Pruebas de Aceptación	18
4.1. Historias de Usuarios	29
4.2. Historia de Usuario Creación de una Jugada	29
4.3. Historia de usuario Visualizar una Jugada	30
4.4. Historia de Usuario Editar una Jugada	30
4.5. Tareas de ingeniera	31
4.6. Tarea de ingeniería 1 para historia de usuario 1	32
4.7. Tarea de ingeniería 2 para historia de usuario 1 y 3	32
4.8. Tarea de ingeniería 3 para historia de usuario 1	32
4.9. Tarea de ingeniería 4 para historia de usuario 1 y 3	33
4.10. Tarea de ingeniería 5 para historia de usuario 1 y 3	33
4.11. Tarea de ingeniería 6 para historia de usuario 1 y 3	33
4.12. Tarea de ingeniería 7 para historia de usuario 1 y 3	34
4.13. Tarea de ingeniería 8 para historia de usuario 2	34
4.14. Tarea de ingeniería 9 para historia de usuario 2	34
4.15. Tarea de ingeniería 10 para historia de usuario 2 y 3	35
4.16. Tarea de ingeniería 11 para historia de usuario 2 y 3	35
4.17. Tarea de ingeniería 12 para historia de usuario 2	35
4.18. Tarea de ingeniería 13 para historia de usuario 3	36
4.19. Tarea de ingeniería 14 para historia de usuario 3	36
4.20. Tarea de ingeniería 15 para historia de usuario 3	36

4.21. Pruebas de Aceptación	47
4.22. Prueba de Aceptación 1	48
4.23. Prueba de Aceptación 2	51
4.24. Prueba de Aceptación 3	53
4.25. Prueba de Aceptación 4	55

www.bdigital.ula.ve

Capítulo 1

Introducción

1.1. Antecedentes

Pacheco y Márquez (2010) en su investigación “*Capacitación a los Entrenadores de la Escuela de Béisbol Menor del Municipio Nirgua, a través de un Programa Teórico Practico Para Mejorar el Desarrollo de las Estrategias Metodológicas*” su enfoque se basó en conocer y mejorar el nivel de preparación y capacitación profesional de los entrenadores de esta escuela deportiva, mediante el uso de entrevistas con los profesores de la escuela deportiva se apreciaron falencias a la hora de impartir nuevas estrategias metodológicas a los estudiantes de la escuela así como la falta de apoyo tecnológico para poder transmitir el conocimientos técnico-táctico.

De una manera similar, Foster y García (2014) en su trabajo “*Jornada de formación permanente sobre la metodología del entrenamiento deportivo aplicada al béisbol para los entrenadores de la escuela de béisbol menor ‘los chamos’ ubicada en la comunidad los próceres del municipio Naguanagua-edo.Carabobo*” donde el objetivo de la investigación

era el de proponer una jornada de formación sobre la metodología de entrenamiento para los entrenadores de la escuela, mediante el uso de la entrevista a varios de los profesores que imparten las clases deportivas junto con la observación reiterada de las prácticas llegaron a la conclusión de que los profesores carecen de estrategias, metodologías y recursos tecnológicos para impartir las prácticas correctas tanto técnicas como tácticas del deporte a la gran cantidad de estudiantes que hacen vida en esta escuela deportiva, esta problemática conlleva a que los estudiantes no realicen los fundamentos técnicos-tácticos de una manera adecuada en la práctica del deporte. Estos últimos trabajos mencionados son una muestra de que en el ámbito nacional una gran parte de las escuelas deportivas no optan por el uso de recursos o herramientas tecnológicas dentro de sus metodologías para impartir un mejor aprendizaje técnico-táctico y que este aprendizaje a su vez sea de mayor entendimiento para sus estudiantes.

Por otro lado, para Barbero et. al. (2006) en su trabajo titulado “*Cuantificación de la Carga en Fútbol: Análisis de un Juego en Espacio Reducido*” uno de los objetivos de su investigación es el de ver la posibilidad de aplicar nuevas herramientas tecnológicas para la monitorización de los entrenamientos de fútbol. Los resultados que se obtuvieron de esa investigación certificaron la posibilidad de poder cuantificar la carga deportiva mediante herramientas tecnológicas lo cual permitió una monitorización precisa y exhaustiva mejorando tanto el rendimiento como el aprendizaje táctico de los jugadores. El trabajo de Barbero et. al. (2006) sirvió como fuente de información para afirmar que herramientas tecnológicas bien aplicadas en la enseñanza deportiva pueden generar un aporte significativo y positivo en el aprendizaje de los jugadores, así como en la preparación por parte de los técnicos o profesores encargados.

En un contexto diferente, Ulloa (2014) en su trabajo “*Estudio de metodologías para estandarizar el desarrollo de software en el departamento de informática en la Pastoral social caritas de la diócesis de Ambato*” marca como objetivo elaborar una guía que mejore el desarrollo de software dentro de la empresa donde realizo la investigación. Dentro de las conclusiones del trabajo se destaca que los procesos utilizados en el departamento de sistemas de la Pastoral para el desarrollo de software no son los adecuados ya que se invierte mayor tiempo en su realización en gran parte por su extensa documentación, a su vez, destaca que el uso de la metodología ágil Extreme Programming (XP) mejorará la eficiencia y calidad de desarrollo de software dentro del departamento, ya que esta metodología ayuda en gran medida la disminución de errores por la comunicación continua que se tiene con el cliente durante todo el desarrollo, además de que la metodología XP cuenta con grandes ventajas como su programación organizada, pruebas continuas y documentación simplificada. En base a las comparativas sobre las diferentes metodologías ágiles que realiza Ulloa en su trabajo y observando sus conclusiones se demuestra que la metodología XP es una de las más útiles y eficientes para el desarrollo de software en lapsos de tiempo no muy amplios, por lo que se considerará a esta metodología la más apropiada para ser adoptada durante este proyecto de grado para la elaboración del software.

1.2 Planteamiento del Problema

El uso de la tecnología se va haciendo presente de manera cotidiana en cada aspecto de nuestras vidas, desde el desarrollo de una pequeña actividad hasta la actividad más compleja puede estar envuelta con el uso de nuevas tecnologías. Es por ello que en campos como el aprendizaje o la enseñanza necesitan con constante recurrencia innovar materiales didácticos que vayan a la par de las tecnologías y de los tiempos en los cuales vivimos.

Izquierdo y González (2006) en su trabajo “*Tecnologías y aplicaciones en la evaluación de la fuerza*” exponen que en el ámbito deportivo cualquier detalle por pequeño que parezca, puede marcar una gran diferencia en el resultado final, es por ello que los métodos de enseñanza y datos que puedan brindar las nuevas tecnologías pueden significar una ventaja competitiva dependiendo de cómo se utilicen. La variedad de métodos de entrenamiento, métodos de aprendizaje y datos que pueden obtenerse a partir de nuevas tecnologías dan paso a una nueva etapa donde los deportistas y entrenadores pueden aprovechar estas nuevas herramientas tecnológicas para mejorar habilidades, entrenamientos y resultados.

Las tecnologías han aportado muchas facilidades y mejoras en lo que respecta al rendimiento táctico en el fútbol, donde se puede ver una constante evolución en software y herramientas tecnológicas aplicadas a este deporte. Se pueden encontrar herramientas que facilitan en gran medida el trabajo tanto de análisis como preparación táctica. Herramientas como *Wyscout*, que es una plataforma con una vasta cantidad de información del fútbol actual para realizar scouting, ver y analizar partidos, analizar jugadas o ver estadísticas de jugadores. También se tienen software como *Longomatch* o *Tactitalpad*, los cuales permiten con facilidad crear paneles de análisis sobre videos, realizar valoraciones tácticas, crear sistemas de juegos, preparar tácticas o sistemas de entrenamiento, entre otras más funcionalidades.

Un aspecto desfavorable en el uso de los software y herramientas mencionadas anteriormente, igual que la gran mayoría de las que se pueden encontrar en el mercado con estas características, es que son de una suscripción paga o no cuentan con una forma de integrarse directamente con las plataformas digitales de los clubes que requieren hacer uso de algunas de esas funcionalidades dentro de sus plataformas. Esta problemática

puede reflejarse en la organización de la Academia Emeritense FC, donde no se cuenta con una herramienta tecnológica propia que permita elaborar a la directiva deportiva del club un plan de trabajo táctico y de entrenamiento de manera conjunta para los entrenadores de las diferentes categorías presentes en esta organización, y que a su vez, permita a sus entrenadores un espacio individual donde plasmar a todos sus ideas tácticas, probarlas y guardarlas de manera tal que los del entorno puedan visualizarlas dentro de la plataforma en cualquier momento y lugar para estudiarlas sin la necesidad de estar en las instalaciones deportivas junto con el entrenador para tener ese aprendizaje táctico.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un software de análisis e implementación táctica para ser utilizado por entrenadores deportivos donde puedan crear, simular y guardar sistemas de juego, de entrenamiento y tácticos con la finalidad de luego ser presentados a los miembros de la plataforma digital del club de futbol Academia Emeritense FC.

1.3.2. Objetivos Específicos

- Identificar los métodos y estrategias que se utilizan para realizar la preparación táctica en el club de futbol Academia Emeritense FC.
- Planificar junto a la organización Academia Emeritense FC las funcionalidades necesarias a implementar en el software para la preparación táctica.

- Elaborar un software con las funcionalidades para emular y mejorar las formas en las que se imparte la preparación táctica en la organización Academia Emeritense FC.

1.4. Justificación

En los últimos tiempos la manera en la cual se realiza cualquier actividad está fuertemente influenciada por las nuevas tecnologías que surgen cada día, la preparación táctica dentro del deporte no escapa de este hecho y es por ello que se necesita de actualizar las formas en las cuales se imparte este aprendizaje táctico dentro las Instituciones deportivas, no basta únicamente con la utilización de metodologías como las charlas tácticas por parte de los entrenadores o la explicación que puedan dar en la pizarra estos, se necesita elevar ese nivel de enseñanza a una forma más didáctica y que no se limite a que ambas partes estén presentes en el mismo espacio para que se pueda realizar.

El aporte que brinda el desarrollo de un software de análisis e implementación táctica deportiva es que los entrenadores pueden hacer uso de diferentes elementos para elaborar y planificar jugadas, sistemas tácticos, sistemas de entrenamiento y otras actividades con la finalidad de que estos sean vistos por los estudiantes desde otro dispositivo en cualquier momento y las veces que se requiera, logrando un mayor entendimiento de este aprendizaje táctico.

Utilizar un software de estas características resolvería otra problemática presente en la organización Academia Emeritense FC la cual es poder sistematizar un plan de trabajo táctico y entrenamiento único para todos los entrenadores de las diferentes categorías del

club y a su vez donde cada entrenador tenga la libertad de poder crear y configurar jugadas o sistemas de juego para que luego sus jugadores las puedan observar dentro de la plataforma digital.

1.5. Propósito

La finalidad de esta tesis es desarrollar un software que sirva como una herramienta de apoyo en la parte técnica para los entrenadores de futbol con las funcionalidades básicas dentro de una plataforma digital. Que en un futuro pueda incorporarse a este software funcionalidades más complejas y nuevas características para ampliar su rango de uso a más de un deporte o disciplina deportiva.

www.bdigital.ula.ve

Capítulo 2

Marco Teórico

Para el correcto entendimiento del desarrollo de este proyecto de grado, en este capítulo se presentan las bases teóricas donde se definirán conceptos del ámbito deportivo relacionados directamente con las funcionalidades que debe tener la herramienta tecnológica a elaborar y posterior a ellos se definirán conceptos para entender las tecnológicas y metodologías utilizadas en los siguientes capítulos.

2.1. Bases Teóricas

2.1.1. Entrenamiento Deportivo

Dietrich et al. (2001) describen al entrenamiento deportivo como un “proceso completo de actividades, dirigido al desarrollo planificado de ciertos estados de rendimiento deportivo y a su exhibición en situaciones de verificación deportiva, especialmente en la actividad competitiva”.

Por otra parte, según la concepción de Platonov (1999) el entrenamiento deportivo lo define como un concepto que reúne todas las medidas del proceso para aumentar el rendimiento deportivo. En todos los ámbitos de competencia es necesario seguir un entrenamiento planificado, pero sobre todo en lo que concierne al deporte, pues este va a comprender un conjunto de tareas que aseguran buena salud, educación, desarrollo físico armónico, amplio dominio técnico-táctico y alto nivel de desarrollo de las capacidades específicas.

2.1.1.1. Sistema de Entrenamiento

Rodríguez y Moreno (1996) definen a los sistemas de entrenamiento como “el conjunto de métodos, actividades y ejercicios que ordenados de forma adecuada sirven para desarrollar la condición física del deportista”. Para que un atleta pueda considerarse con expectativas reales de triunfo en su área, éste debe seguir un sistema de preparación específico, por ello existe una diversidad de tipos de entrenamientos para atletas de deportes individuales y de conjunto. Entre estos sistemas, se encuentran aquellos que desarrollan su aptitud muscular, velocidad, movilidad entre otras.

Por su parte, Dietrich et al. (2001), exponen que el entrenamiento se ha de concebir ante todo como un sistema global con la exigencia de una sistematización y planificación, por tanto, un *sistema de entrenamiento* debe ir orientado a un objetivo común, este consta de un conjunto de procesos individuales y de medidas que, de acuerdo con las leyes de la evolución del rendimiento deportivo, interactúan entre sí en relación cambiante y que dependen dinámicamente unos de otros.

2.1.2. Método de Entrenamiento

Zintl (1991) describe el método de entrenamiento como un procedimiento programado que determina los contenidos, medios y cargas de entrenamiento en función de los objetivos. Los métodos de entrenamiento representan las pautas específicas de trabajo que se utilizan para la preparación de un deportista. Cuando esta pauta se aplica a una cualidad física concreta, ya no se trata de un sistema de entrenamiento, sino de un método.

2.1.3. Táctica Deportiva

La táctica deportiva se puede definir como el comportamiento racional, regulado sobre la propia capacidad de rendimiento del deportista y sobre la del adversario, así como sobre las condiciones exteriores en un encuentro deportivo que sea individual o en equipo (Weineck, 2005).

La táctica en el deporte se va construyendo desde el técnico o cuerpo técnico hacia los jugadores, mediante la preparación de videos, estadísticas, redacción de informes, presentaciones, charlas, al tiempo que analizan puntos fuertes y débiles del equipo y su adversario.

Para Weineck, la táctica deportiva se concibe sobre tres pilares:

- a. *Capacidades cognitivas*: La formación del pensamiento táctico debe ser progresiva, al tiempo que involucra aspectos tanto teóricos como prácticos; además de la variabilidad de las situaciones propuestas es clave para enriquecer el aprendizaje.

- b. *Habilidades técnicas*: El dominio de la técnica supone una ventaja táctica en tanto economía de esfuerzo y recursos. La forma deportiva condiciona la técnica y táctica. A mayor estado físico, mayores probabilidades de afrontar problemas tácticos.
- c. *Capacidad psico-físicas*: Las cualidades emotivas y de decisión o voluntad se van forjando a través de la experiencia. Es probable que sean los únicos elementos que no puedan trabajarse de otra manera.

2.1.4. Sistemas de Juego

Espar (2004) define el sistema de juego como el conjunto de normas que rigen las acciones individuales y colectivas, cuyo objetivo es obtener el máximo rendimiento de todo el grupo.

Un sistema de juego parte de la táctica y puede considerarse como el funcionamiento del equipo para conseguir ventajas numéricas en todas las zonas de campo, con la intención de superar al contrario tanto en la fase ofensiva del equipo como en la fase defensiva.

2.1.5. Metodología Desarrollo de Software

Existe una variedad de metodologías para el desarrollo de softwares, cada una con características distintas o similares que no necesariamente son adecuadas para todo tipo de proyectos; cada una de ellas se adaptan a ciertos tipos de proyectos, con variaciones técnicas, organizacionales o según el tipo de equipo de trabajo.

2.1.5.1. Metodologías Tradicionales

Este tipo de metodologías se caracteriza por tener una alta estructuración y planeación de las actividades del proceso de desarrollo a efectuar desde el inicio del proyecto. Las metodologías bajo este enfoque requieren una definición clara del proceso de desarrollo de sistemas, se basan en una documentación comprensiva que se traduce en actividades más predictivas y eficientes. Las metodologías tradicionales pueden ser exitosas si son utilizadas para desarrollar sistemas donde los requerimientos son claros desde el inicio.

2.1.5.2. Metodologías Ágiles

Se pueden definir como metodologías que poseen un proceso de desarrollo incremental de fácil comprensión y adaptación a los cambios, donde existe una cooperación y comunicación constante entre el cliente y los desarrolladores (Geambasu, et al., 2011)

Estas metodologías ágiles se establecieron como una alternativa a los procesos tradicionales de desarrollo de software que se caracterizan por ser más rígidos y cautivos de condiciones de uso severas, las cuales son plasmadas en una documentación extensa a generar en cada una de las etapas de desarrollo.

2.1.5.3. Metodología Extreme Programming (XP)

La Programación Extrema o Extreme Programming, es una metodología para desarrollo de software formulada por Kent Beck y es considerada como una de las más destacadas metodologías ágiles de desarrollo de software. La programación extrema al igual que las demás metodologías ágiles se diferencia de los métodos tradicionales principalmente en que presenta más énfasis en la adaptabilidad que en la previsibilidad. (Bautista, 2012)

Esta metodología ágil se basa en la simplicidad, la comunicación y la realimentación del código desarrollado. Está diseñada para proyectos que requieren un grupo de programadores pequeño, dónde la comunicación sea más fácil y factible que en grupos de desarrollo grandes, pues la comunicación entre desarrolladores, programadores, jefes de proyecto y los clientes es de suma importancia.

2.1.5.3.1. Objetivos

- La satisfacción del cliente.
- Potenciar el trabajo en grupo.
- Minimizar el riesgo actuando sobre las variables del proyecto: costo, tiempo, calidad y alcance.

www.bdigital.ula.ve

2.1.5.3.2. Características

Para Meléndez et al. (2016) las características de la programación extrema se pueden resumir de la siguiente manera:

- Metodología basada en prueba y error para obtener un software que funcione realmente.
- Está orientada hacia quien produce y usa software (el cliente participa muy activamente).
- Reduce el coste del cambio en todas las etapas del ciclo de vida del sistema
- Combina las que han demostrado ser las mejores prácticas para desarrollar software, y las lleva al extremo
- Los requisitos pueden cambiar.

- Grupo pequeño y muy integrado (2-10 personas).

2.1.5.3.3. Fases

▪ **Planificación**

La primera fase consiste en identificar y elaborar las historias de usuario, en las cuales se detallan los requisitos específicos del software que se va a desarrollar. Estas las historias de usuario incluyen las necesidades, tiempos y estimaciones de los clientes para poder planificar la gestión de acuerdo a las prioridades. Las historias se descomponen en mini versiones que permitan hacer un seguimiento periódicos

▪ **Diseño**

Se elaboran los prototipos y se hacen códigos sencillos para probar la efectividad del programa o funcionalidades, la idea en esta fase es que el código que se cree tenga suficiente capacidad para cubrir con la historia del usuario, es decir, sus necesidades. El diseño debe ser flexible e integrado para que al momento de ejecutar la prueba pueda pasar a la siguiente fase.

▪ **Codificación**

En esta fase de la metodología XP se genera un código universal y en equipo con la finalidad de que todo el equipo pueda avanzar de manera simultánea, al tener un conocimiento completo del estado del proyecto. De esta manera, se busca evitar que la creación esté dirigida por un solo programador

- **Pruebas**

los códigos que se generan a lo largo de la fase de codificación son sometidos a pruebas para corregir fallas y adaptar nuevas mejoras. Al tratarse de proyectos a corto plazo, la realización constante de pruebas sobre los códigos realizado es un punto clave. Además, el propio cliente puede realizar pruebas, proponer nuevas pruebas e ir validando las mini-versiones que se van obteniendo.

- **Lanzamiento**

Al llegar a esta fase es porque los códigos han pasado la prueba en cada fase y se tiene un software útil que responde a las necesidades del cliente.

www.bdigital.ula.ve

2.1.5.3.4. Herramientas

- **Historias de Usuario**

Representan una breve descripción del comportamiento del software, estas se realizan por cada característica principal que cuente el software y son utilizadas para cumplir estimaciones de tiempo y el plan de iteraciones. Suelen reemplazar un gran documento de requisitos y presiden la creación de las pruebas de aceptación. Cada historia de usuario debe ser lo suficientemente comprensible y delimitada para que los desarrolladores puedan entenderlas con facilidad e implementarlas en el tiempo estimado.

Cuadro 2.1. Plantilla para Historias de Usuario

Historia de Usuario	
Número: Permite identificar cual historia de usuario se está desarrollando.	Usuario: Persona que utilizará la funcionalidad descrita en la historia de usuario
Nombre historia: : Describe de manera general a una historia de usuario	
Prioridad en negocio: Grado de importancia que el cliente asigna a una historia de usuario. (Alta / Media / Baja)	Riesgo en desarrollo: complejidad que una historia de usuario representa al equipo de desarrollo. (Alta / Media / Baja)
Puntos estimados: Número estimado de semanas que se necesitará para el desarrollo de una historia de usuario	Iteración asignada: Número de iteración, en que se implementara una historia de usuario.
Descripción: Información detallada de una historia de usuario	

▪ **Tareas De Ingeniería - Task Card**

Las tareas de Ingeniería describen cuales actividades se realizarán en cada historia de usuario, así mismo las tareas de ingeniería se vinculan directamente al desarrollador, ya que permite tener un primer acercamiento con el código y las funcionalidades a desarrollar.

Cuadro 2.2. *Plantilla para Tareas de Ingeniera*

Tarea de Ingeniera	
Número: Permite identificar cual tarea de ingenieria se está desarrollando.	Numero de Historia de Usuario: Número asignado de la historia correspondiente a esta tarea.
Nombre Tarea: Describe de manera general a una tarea de ingeniería.	
Tipo de Tarea: Tipo al que corresponde la tarea de ingeniería.	Puntos Estimados: Número de días estimado que se necesitará para el desarrollo de una tarea de ingeniería
Descripción: : Información detallada de la tarea de ingeniería..	

www.bdigital.ula.ve

▪ **Pruebas de Aceptación**

Para Chiluisa y Loarte (2014) Las pruebas de aceptación son sumamente importantes para el éxito de una iteración y el comienzo de la siguiente, mediante estas el cliente

puede conocer el avance en el desarrollo del sistema y a los programadores lo que les resta por desarrollar. Además, permite una retroalimentación para el desarrollo de las próximas historias de usuarios a ser entregadas. Estas son comúnmente llamadas pruebas del cliente, por lo que son realizadas por el encargado de verificar si las historias de usuarios de cada iteración cumplen con la funcionalidad esperada.

Cuadro 2.3. *Plantilla para Pruebas de Aceptación*

Prueba de Aceptación	
Número: número que permite identificar la prueba de aceptación.	Historia de Usuario: Número que identifica la historia de usuario vinculada a la prueba de aceptación.
Nombre Prueba Aceptación: Nombre que indica de manera general la descripción de la prueba de aceptación	
Condiciones de Ejecución: Condiciones previas que deben cumplirse para realizar la prueba de aceptación.	
Pasos de Ejecución: Pasos que siguen los usuarios para probar la funcionalidad de la historia de usuario.	
Resultado Esperado: Respuesta del sistema que el cliente espera, después de haber ejecutado una funcionalidad.	
Evaluación de la Prueba: Nivel de satisfacción sobre la respuesta de la funcionalidad del sistema o software.	

2.1.6. JavaScript

JavaScript es un lenguaje de programación interpretado de alto nivel que apareció por primera vez en 1995. Se caracteriza por ser un lenguaje funcional multi-paradigma, orientado a objetos, imperativo y dinámico.

Este lenguaje de programación se utiliza principalmente del lado del cliente, implementado como parte de un navegador web lo que permite mejoras en la interfaz de usuario y páginas web dinámicas, además permite ejecutar rutinas automatizando ciertos procesos dentro de la maquetación web (Flanagan, 1998)

2.1.7. HTML

HTML (*HyperText Markup Language*) es un Lenguaje de Marcado de Hipertexto, el cual se utiliza para estructurar y desplegar una página web y sus contenidos. Este lenguaje permite describir hipertexto, texto presentado de forma estructurada y agradable, generar archivos pequeños, despliegue rápido, ser un lenguaje de fácil aprendizaje, además que es un lenguaje que admiten todos los navegadores web.

www.bdigital.ula.ve

2.1.8. CSS

CSS (*Cascading Style Sheets*) es un lenguaje de diseño gráfico que permite definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Este lenguaje permite a los desarrolladores web regular varios elementos de estilo y funcionalidades, como el diseño, el color, las fuentes y por ende el formato y la visualización de documentos HTML.

2.1.9. Framework

Es una aplicación o software reutilizable, que puede ser especializada para la producción de nuevas aplicaciones concretas y específicas. El framework describe los objetos que

componen el sistema y cómo éstos interactúan, sus interfaces y el flujo de control entre ellos (Johanson y Foote, 1988)

Los frameworks se diseñan con el objetivo de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional.

2.1.9.1. Konva

Konva es un framework Javascript de lienzo que amplía las funciones 2D al hacer que tanto aplicaciones de escritorio web como aplicaciones móviles sean más interactivas. Este framework permite animaciones de alto rendimiento, transiciones, anidamiento de nodos, capas, filtros, almacenamiento en cache, anejo de eventos para diferentes aplicaciones.

2.1.10. Wireframe

Es un término que usa para la representación esquemática de cada una de las vistas más significativas que van a componer un producto y reflejan las funcionalidades que estarán presentes. Además, permite visualizar la distribución los contenidos dentro de cada vista por lo que se suelen utilizar para el diseño de cualquier tipo de interfaz. Los software o herramientas más utilizados para la elaboración de wireframes están: InVision, Balsamiq Mockup, Lucidchart entre otros.

2.1.11. Bases de Datos

Las bases de datos son un almacén de información actualizable de una aplicación, que oculta del usuario los aspectos físicos del almacenamiento y la representación de la información. La información que se almacena en una base de datos es accesible a un nivel lógico sin necesidad de involucrar los conceptos físicos de su implementación. Los modelos de bases de datos se pueden clasificar según su estructura lógica y el modo de almacenar, organizar y manipular los datos (Rishe, 1992).

2.1.11.1. Bases de Datos Relacionales

Rouse (2022) define a una base de datos relacional como “una colección de datos organizados en un grupo de tablas formalmente descritas, las cuales pueden ser accedidas de diferentes formas”. Algunas bases de datos relacionales conocidas son: MySQL, PostgreSQL, MariaDB entre otras.

2.1.11.2. Bases de Datos No Relacionales

Una *base de datos no relacional* es aquella que no utiliza un esquema tabular de filas y columnas como se concibe en la mayoría de los sistemas de base de datos más tradicionales o base de datos relacionales. En su lugar, las bases de datos no relacionales utilizan un modelo de almacenamiento que está optimizado para los requisitos específicos del tipo de datos que se almacena (Tejada, 2022). Algunas bases de datos no relacionales conocidas son MongoDB, Cassandra o BigTable.

2.1.11.2.1. MongoDB

Es una base de datos no relacional de código abierto. Puede procesar datos estructurados, semiestructurados y no estructurados, es muy flexible y permite combinar y almacenar múltiples tipos de datos. También almacena y maneja mayores cantidades de datos que las bases de datos relacionales tradicionales. MongoDB usa un formato de almacenamiento de documentos llamado BSON, que es una forma binaria de JSON que puede acomodar más tipos de datos.

www.bdigital.ula.ve

Capítulo 3

Marco Metodológico

Este capítulo describe los aspectos vinculados con el tipo y diseño de la investigación que se aplica en el presente trabajo de grado, así como la metodología y las fases utilizadas para el desarrollar el software de análisis e implementación táctica.

3.1. Diseño de Investigación

El diseño de la presente investigación es de campo, debido a la necesidad de recolectar la información necesaria directamente de los actores implicados para cumplir con el objetivo principal de este proyecto de Grado.

3.2. Tipo de Investigación

Esta investigación se considera de tipo aplicada, ya que se plantea como objetivo general el desarrollar un software de análisis e implementación táctica deportiva para resolver una necesidad presente en la Academia Emeritense.

3.3. Nivel de Investigación

La investigación se puede catalogar en los siguientes niveles de investigación:

Exploratorio:

Ya que fue necesario comprender la realidad del problema que se quiere dar solución con esta investigación, además de que no existe un cuerpo teórico suficiente en el cual se pueda apoyar para desarrollar de la mejor manera la solución a la necesidad presentada en este proyecto de grado.

Aplicativo:

Debido a que se cumplió con el objetivo de la investigación el cual era brindar una solución palpable a la problemática o necesidad mediante el desarrollo de un software.

3.4 Metodología a Utilizar

Para el desarrollo del software se utilizó la metodología XP (Extreme Programming). Se eligió esta metodología Ágil ya que es una de las más eficientes en cuanto a desarrollo de software en periodos cortos de tiempo. Además, varias de las características de esta

metodología como la adaptabilidad, flexibilidad o comunicación constante con el cliente se consideraron necesarias para poder llevar acabo el desarrollo del software de análisis e implementación táctica deportiva.

Las fases de desarrollo que se utilizaron de esta metodología ágil y que se explicaran a detalle en el siguiente capítulo son las siguiente:

- Planificación
- Diseño
- Codificación
- Pruebas

www.bdigital.ula.ve

Capítulo 4

Desarrollo del Proyecto

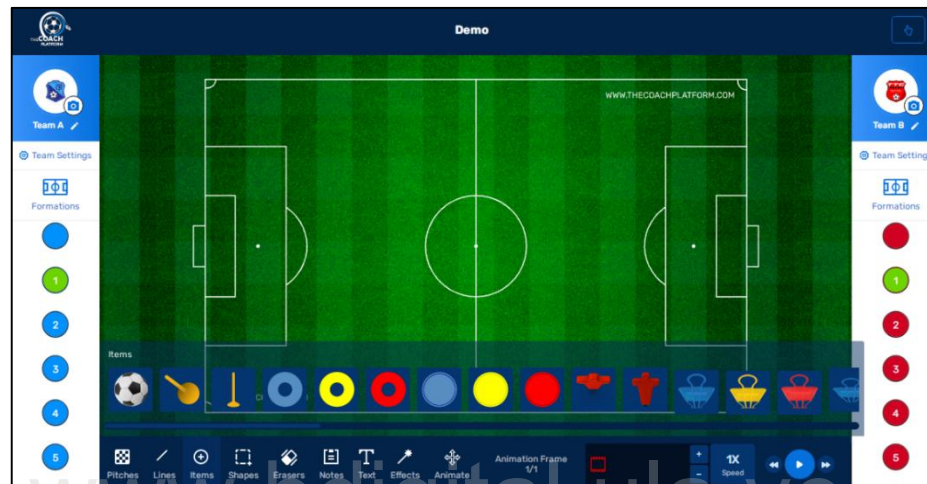
Durante este capítulo se llevará a cabo el desarrollo del software análisis e implementación táctica deportiva, en donde se tomará como referencia a la metodología ágil XP y las fases que componen a esta metodología.

4.1 Planificación

A lo largo de esta primera fase se realizaron diversas reuniones y entrevistas con la dirección deportiva de la Academia Emeritense en donde se pudo comprender cuales son los métodos, estrategias y herramientas que están utilizando actualmente para llevar a cabo la planificación táctica dentro de la Academia. En este aspecto táctico-deportivo la dirección deportiva expuso como realizan sus sistemas de entrenamiento deportivos, movimientos y sistemas tácticos, jugadas de estrategia (a este grupo de actividades se hará futuras referencias con el término ‘Jugadas’) e hicieron mención de los principales softwares deportivos de suscripción paga que han utilizado en esta institución deportiva Se tomaron como referencia

dos software para el diseño e implementación de las actividades ya mencionadas, el primero de ellos es el software web *TheCoachPlatform* y el software de escritorio *TacticalPad*.

Figura 4.1. Vista Software profesional *TheCoachPlatform*



Nota. Tomado del software *TheCoachPlatform*

Figura 4.2. Vista Software profesional *TacticalPad*



Nota. Tomado del software *TacticalPad*

Durante estas reuniones se lograron definir los requerimientos necesarios que tendrá el software, donde se tomó en consideración la prioridad y tiempo que se tiene para el desarrollo de cada historia de usuario.

El principal requerimiento de este software será un ambiente o entorno de trabajo que permita al usuario crear de una manera sencilla y sin complejidad *Jugadas* donde los elementos utilizados puedan ser personalizables en función a cada técnico y que se puedan almacenar en una base de datos. Otro requerimiento en el software es tener una vista donde se pueda recuperar todas las ‘jugadas’ elaboradas por un usuario, seleccionar una de ellas y poder visualizarla. Por ultimo también se requiere de un ambiente donde el usuario pueda seleccionar una jugada de su propia elaboración y realizar unos cambios sobre este si así lo desea.

www.bdigital.ula.ve

El software estará planeado para una futura integración con la plataforma digital de la Academia, por lo que se partirá de premisa que los usuarios que harán uso de las funcionalidades ya están definidos e implementados.

4.1.1 Historias de Usuario

Las historias de usuarios (HU) que se elaboraron están descritas en un lenguaje común y sencillo, de modo que puedan ser entendidas por los implicados en la planificación y elaboración del software, estas historias de usuario representan los requerimientos que debe cumplir el software y son las siguientes:

Cuadro 4.1. Historias de Usuarios

Historia de Usuarios	
Numero	Nombre
1	Creación de una <i>Jugada</i>
2	Visualizar <i>Jugadas</i>
3	Editar una <i>Jugada</i>

A continuación, en los cuadros 4.2- 4.4 se muestra con detalle las historias de usuario que se usaron para el desarrollo del software.

Cuadro 4.2. Historia de Usuario Creación de una Jugada

Historia de Usuario	
Número: 1	Usuario: Técnicos
Nombre historia: Creación de una Jugada.	
Prioridad en negocio: Alta (Alta / Media / Baja)	Riesgo en desarrollo: Bajo (Alta / Media / Baja)
Puntos estimados: 1	Iteración asignada: 1
Descripción: El usuario puede realizar las siguientes tareas en esta funcionalidad: • Ingresar un título y descripción de la jugada a crear. • Seleccionar una categoría y situación específica de jugada. • Seleccionar el ambiente el ambiente de trabajo donde se elabora la jugada. • Ingresar ítems al ambiente de trabajo en donde se puedan ubicar en la posición deseada. • Crear una secuencia animada con los ítems ingresados en el ambiente de trabajo. • Guardar la información general de la jugada y la secuencia de movimientos de los ítems	

Cuadro 4.3. Historia de usuario Visualizar una Jugada

Historia de Usuario	
Número: 2	Usuario: Técnicos y estudiantes
Nombre historia: Visualizar Jugadas	
Prioridad en negocio: Alta (Alta / Media / Baja)	Riesgo en desarrollo: Bajo (Alta / Media / Baja)
Puntos estimados: 1	Iteración asignada: 2
Descripción: El usuario debe seleccionar el nombre de un usuario para cargar la lista de jugadas vinculadas a ese usuario. Luego debe escoger una las Jugadas de la lista y poder visualizar la jugada seleccionada.	

Cuadro 4.4. Historia de Usuario Editar una Jugada

Historia de Usuario	
Número: 3	Usuario: Técnicos
Nombre historia: Editar una Jugada	
Prioridad en negocio: Media (Alta / Media / Baja)	Riesgo en desarrollo: Medio (Alta / Media / Baja)
Puntos estimados: 1	Iteración asignada: 3
Descripción: El usuario puede seleccionar una Jugada entre su lista de Jugadas, luego debe visualizar la jugada seleccionada teniendo dos opciones: • Modificar la información general junto la secuencia de movimientos relacionada a la jugada seleccionada. Luego guardar la jugada. • Eliminar la Jugada	

4.1.2 Tareas de Ingeniería

En el cuadro 4.5 se muestran de manera general las correspondientes tareas de ingenierías y en los cuadros 4.6 – 4.20 se detallan cada una de ellas.

Cuadro 4.5. *Tareas de ingeniera*

Tareas de Ingeniería		
Numero Tarea	Numero de Historia Usuario	Nombre de la Tarea
1	1	Diseño interfaz para crear nueva Jugada
2	1, 3	Captura de información general de una Jugada
3	1	Instanciar un ambiente de trabajo
4	1, 3	Agregar y eliminar ítems en el ambiente de trabajo
5	1, 3	Personalizar un ítem
6	1, 3	Generar secuencia de movimientos de los Ítems dentro del ambiente de trabajo.
7	1, 3	Guardar información general y la secuencia de movimientos de una Jugada.
8	2	Diseñar interfaz para Visualizar una Jugada
9	2	Listar usuarios que tienen creada alguna Jugada.
10	2, 3	Listar Jugadas de un usuario
11	2,3	Cargar información general de una jugada seleccionada
12	2	Cargar ambiente de trabajo de una jugada seleccionada (para visualizar)
13	3	Diseñar interfaz para Editar una Jugada
14	3	Cargar ambiente de trabajo de una jugada seleccionada (para edición)
15	3	Eliminar una Jugada seleccionada

Cuadro 4.6. Tarea de ingeniería 1 para historia de usuario 1

Tarea de Ingeniera	
Número: 1	Numero de Historia de Usuario: 1
Nombre Tarea: Diseñar interfaz para crear nueva Jugada	
Tipo de Tarea: Tipo al que corresponde la tarea de ingeniería	Puntos Estimados: 1
Descripción: Diseñar una interfaz que permita capturar los valores generales de una nueva jugada y además generar un ambiente de trabajo donde se puedan agregar ítems y crear una secuencia de movimientos deseada.	

Cuadro 4.7. Tarea de ingeniería 2 para historia de usuario 1 y 3

Tarea de Ingeniera	
Número: 2	Numero de Historia de Usuario: 1, 3
Nombre Tarea: Captura de información general de una Juagada	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Descripción: se requiere capturar valores ingresados y seleccionados por el usuario de una nueva Jugada. Estos valores son: Titulo, descripción, situación, categoría y ambiente de trabajo.	

Cuadro 4.8. Tarea de ingeniería 3 para historia de usuario 1

Tarea de Ingeniera	
Número: 3	Numero de Historia de Usuario: 1
Nombre Tarea: Instanciar un ambiente de trabajo	
Tipo de Tarea: Desarrollo	Puntos Estimados: Medio
Descripción: Se necesita instanciar el ambiente de trabajo en donde se ingresarán los ítems y sobre este ambiente se creara la secuencia de movimientos de dichos ítems.	

Cuadro 4.9. Tarea de ingeniería 4 para historia de usuario 1 y 3

Tarea de Ingeniera	
Número: 4	Numero de Historia de Usuario: 1, 3
Nombre Tarea: Agregar y eliminar ítems en el ambiente de trabajo	
Tipo de Tarea: Desarrollo	Puntos Estimados: Medio
Descripción: Se requiere agregar o eliminar cualquier ítem seleccionado dentro del ambiente de trabajo.	

Cuadro 4.10. Tarea de ingeniería 5 para historia de usuario 1 y 3

Tarea de Ingeniera	
Número: 5	Numero de Historia de Usuario: 1, 3
Nombre Tarea: Personalizar un ítem	
Tipo de Tarea: Desarrollo	Puntos Estimados: Medio
Descripción: Se necesita la función de personalizar un ítem según el criterio del usuario, donde este pueda dar un nombre o posición de campo al ítem.	

Cuadro 4.11. Tarea de ingeniería 6 para historia de usuario 1 y 3

Tarea de Ingeniera	
Número: 6	Numero de Historia de Usuario: 1, 3
Nombre Tarea: Generar secuencia de movimientos de ítems dentro del ambiente de trabajo	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Descripción: Se necesita elaborar de manera sencilla una secuencia de movimientos de los ítems.que se ingresaron dentro del ambiente de trabajo.	

Cuadro 4.12. Tarea de ingeniería 7 para historia de usuario 1 y 3

Tarea de Ingeniera	
Número: 7	Numero de Historia de Usuario: 1, 3
Nombre Tarea: Guardar información general y la secuencia de movimientos de una Jugada.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Descripción: Se requiere guardar en base de datos los valores seleccionados e ingresados de una Jugada así como la secuencia de movimientos de los ítems.	

Cuadro 4.13. Tarea de ingeniería 8 para historia de usuario 2

Tarea de Ingeniera	
Número: 8	Numero de Historia de Usuario: 2
Nombre Tarea: Diseñar interfaz para visualizar una Jugada	
Tipo de Tarea: Diseño	Puntos Estimados: 2
Descripción: Diseñar una interfaz que permita seleccionar un elemento de una lista de usuario, seleccionar un elemento de una lista de jugadas de un usuario, mostrar un ambiente de trabajo con la jugada seleccionada.	

Cuadro 4.14. Tarea de ingeniería 9 para historia de usuario 2

Tarea de Ingeniera	
Número: 9	Numero de Historia de Usuario: 2
Nombre Tarea: Listar usuarios que tienen creada alguna Jugada	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Descripción: Se necesita listar a todos los usuarios que tengan creada al menos una jugada en la base de datos.	

Cuadro 4.15. Tarea de ingeniería 10 para historia de usuario 2 y 3

Tarea de Ingeniera	
Número: 10	Numero de Historia de Usuario: 2, 3
Nombre Tarea: Listar Jugadas de un usuario	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Descripción: Se necesita listar todas las Jugadas de un usuario seleccionado y poder seleccionar una de estas Jugadas. A esta lista se debe filtrar según dos criterios: Categoría y situación.	

Cuadro 4.16. Tarea de ingeniería 11 para historia de usuario 2 y 3

Tarea de Ingeniera	
Número: 11	Numero de Historia de Usuario: 2, 3
Nombre Tarea: Cargar información general de una jugada seleccionada	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Descripción: Se requiere cargar valores generales de una jugada previamente seleccionada. Estos valores son: Titulo, descripción, categoría y situación.	

Cuadro 4.17. Tarea de ingeniería 12 para historia de usuario 2

Tarea de Ingeniera	
Número: 12	Numero de Historia de Usuario: 2
Nombre Tarea: Cargar ambiente de trabajo de una jugada seleccionada (para visualizar)	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Descripción: Se requiere cargar el ambiente de trabajo de una jugada previamente seleccionada donde solo se permita visualizar los movimientos en este ambiente sin poder editar o modificar algo dentro de él.	

Cuadro 4.18. Tarea de ingeniería 13 para historia de usuario 3

Tarea de Ingeniera	
Número: 13	Numero de Historia de Usuario: 3
Nombre Tarea: Diseñar interfaz para editar una Jugada	
Tipo de Tarea: Diseño	Puntos Estimados: 2
Descripción: Diseñar una interfaz que permita seleccionar un elemento de una lista de jugadas de un usuario, mostrar un ambiente de trabajo con la jugada seleccionada junto a componentes necesarios para la edición del ambiente de trabajo.	

Cuadro 4.19. Tarea de ingeniería 14 para historia de usuario 3

Tarea de Ingeniera	
Número: 14	Numero de Historia de Usuario: 3
Nombre Tarea: Cargar ambiente de trabajo de una jugada seleccionada (para editar)	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Descripción: Se requiere cargar el ambiente de trabajo de una Jugada previamente seleccionada donde solo se pueda modificar o cambiar la secuencia de movimientos de esta Jugada al igual que los ítems dentro del ambiente de trabajo.	

Cuadro 4.20. Tarea de ingeniería 15 para historia de usuario 3

Tarea de Ingeniera	
Número: 15	Numero de Historia de Usuario: 3
Nombre Tarea: Eliminar una Jugada seleccionada.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Descripción: Se necesita eliminar de base de datos una Jugada previamente seleccionada.	

4.2 Diseño

Luego que se definieron las historias de usuario se diseñaron los wireframe o bocetos de cada una de las funcionalidades que contendrá el software. Para el diseño de estos bocetos se utilizó el software de desarrollo *Balsamick mockups*.

4.2.1. Diseño funcionalidad 1

Funcionalidad relacionada con la HU1 '*Creación de Jugada*'. Para esta funcionalidad se deben tener componentes para el ingreso y selección por parte del usuario para la información general de la jugada, estos valores son: *Título*, *descripción*, *situación* y *categoría*. Posterior a ello se debe elegir entre varios ambientes de trabajo de una manera sencilla y por último se debe visualizar un ambiente de trabajo elegido en donde se puedan agregar varios ítems o elementos los cuales se puedan ubicar en diferentes posiciones para poder crear *Jugadas* haciendo uso de botones de acción para la animación y al final poder guardar esta *Jugada* en la base de datos.

Figura 4.3. Boceto Funcionalidad 1.

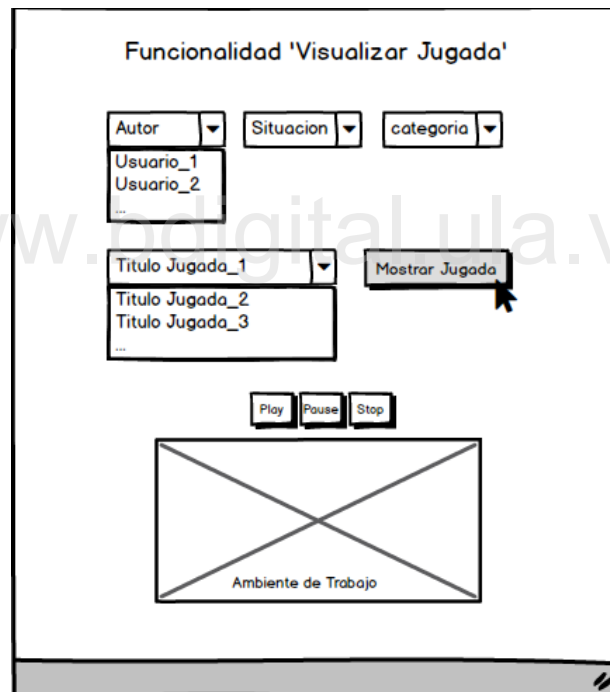
The wireframe diagram, titled 'Funcionalidad Creacion Jugada', illustrates the layout for creating a move. It includes the following elements:

- Input Fields:** 'Titulo de la Jugada' and 'Descripcion de la Jugada' at the top.
- Dropdowns:** 'Categoria Jugada' (with 'Ataque' selected) and 'Situacion Jugada'.
- Environment Selection:** A section labeled 'Selecccion de Ambiente' with three square icons; the third icon is selected with a mouse cursor.
- Action Buttons:** 'Guardar', 'Agregar Movi.', and 'Eliminar Movi.'.
- Animation Controls:** 'Play', 'Pause', and 'Stop' buttons.
- Workspace:** A large rectangular area labeled 'Ambiente de Trabajo' with a large 'X' across it, indicating a placeholder for a visual environment.
- Item Management:** Two columns at the bottom, each with an 'Items' dropdown and an 'Imagen' input field.

4.2.2. Diseño funcionalidad 2

Funcionalidad relacionada a la HU2 '*Visualizar una Jugada*'. Esta funcionalidad debe tener componentes que permitan seleccionar el nombre de un usuario que haya creado previamente una Jugada, posterior a esto se debe poder elegir entre las jugadas del usuario seleccionado teniendo la opción de filtrar estas Jugadas según la categoría o situación. Luego se debe visualizar la Jugada seleccionada.

Figura 4.4. Boceto Funcionalidad 2.



4.2.3. Diseño funcionalidad 3

Funcionalidad relacionada a la HU3 '*Editar una Jugada*'. Esta funcionalidad debe tener componentes que permitan elegir entre las Jugadas de un usuario, con la opción de poder filtrar las Jugadas según la categoría o situación. Al seleccionar la Jugada se debe cargar la

información general (título, descripción, categoría, situación) y el ambiente de trabajo de la misma junto con botones de acción y componentes que permitan agregar o eliminar ítems del ambiente de trabajo, así como para modificar la secuencia de movimiento de los mismos.

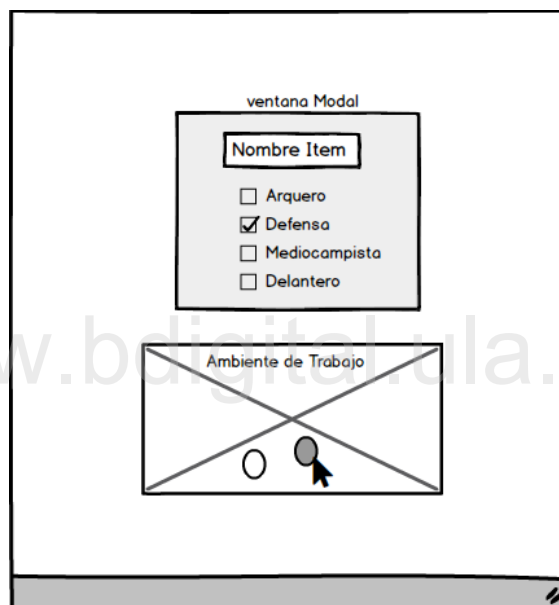
Figura 4.5. Boceto Funcionalidad 3.

The sketch shows a web interface titled "Funcionalidad 'Editar Jugada'". It includes several input fields and buttons. At the top, there are three dropdown menus: "Autor" (with options "Usuario_1", "Usuario_2", "..."), "Situacion", and "categoria". Below these is a dropdown for "Titulo Jugada_1" (with options "Titulo Jugada_2", "Titulo Jugada_3", "...") and a "Mostrar Jugada" button with a mouse cursor. Further down are text inputs for "Titulo Jugada_1" and "Descripcion Jugada_1", followed by two more dropdowns: "Categoria Jugada" and "Situacion Jugada". Below these are four buttons: "Guardar Cambios", "Eliminar Jugada", "Agregar Movimiento", and "Eliminar Movimiento". Underneath are three small buttons: "Play", "Pause", and "Stop". A large rectangular area labeled "Ambiente de Trabajo" is crossed out with a large 'X'. At the bottom, there are two "Items" dropdown menus, each with options "Item_1", "Item_2", and ".....". The entire interface is enclosed in a rectangular frame with a small icon in the bottom right corner.

4.2.4. Diseño funcionalidad 4

Funcionalidad relacionada a la HU1 'Crear una Jugada' y a la HU3 'Editar una Jugada'. Esta funcionalidad debe permitir personalizar los ítems dentro del ambiente de trabajo de alguna Jugada. Debe tener componentes para permitir ingresar valores para el ítem.

Figura 4.6. Boceto Funcionalidad 4.



4.2.5. Selección de Tecnologías

El software será elaborado con tecnologías de fácil integración para que pueda ser incluido en un futuro próximo con la plataforma deportiva de la Academia que está en plena elaboración al momento de la realizar este proyecto de grado. Por lo antes mencionado para la interfaz se usarán las tecnologías HTML, CSS y JavaScript. Para el manejo de los datos en el backend se utilizará el sistema para la gestión de datos MongoDB ya que esta tecnología es la misma en que se elaborara el backend de la plataforma deportiva y así tener una integración más sencilla.

4.3 Codificación

Esta fase llevó a cabo la implementación de las funcionalidades descritas previamente, estas implementaciones se realizaron a lo largo de 3 iteraciones. Debido a que los componentes de las funcionalidades tienen similitud y codependencia, en la primera iteración se implementó la funcionalidad 1, que permite llevar a cabo la creación de una nueva *Jugada*, en la segunda iteración se implementa la funcionalidad 2 la cual se permite seleccionar y mostrar una *Jugada* ya creada. Por último, en la tercera iteración se implementa la funcionalidad 3 que permite seleccionar una *Jugada*, modificar sus ítems y secuencia de movimientos, guardas estos cambios en base de datos o poder eliminar esa *Jugada* seleccionada.

En las funcionalidades para la implementación del ambiente de trabajo se utilizó el framework de lienzo 2D Konva, se usaron las funciones de este framework para instanciar el ambiente de trabajo, para agregar o eliminar ítems, la ubicación de los ítems y para elaborar la secuencia animada de movimientos. Las imágenes de fondo utilizadas para los ambientes de trabajo se tomaron del software de análisis táctico *TacticalPad* debido a que no se terminó la elaboración de estas imágenes de fondo durante el desarrollo de este proyecto de grado.

4.3.1 Implementación Funcionalidad 1

Figura 4.7. Primera parte Implementación Funcionalidad 1.

Crear Jugada

Titulo Nueva Jugada

Descripcion Jugada

Categoria Jugada: Defensa

Situacion: Dinamica

Elige un Campo o Ambiente

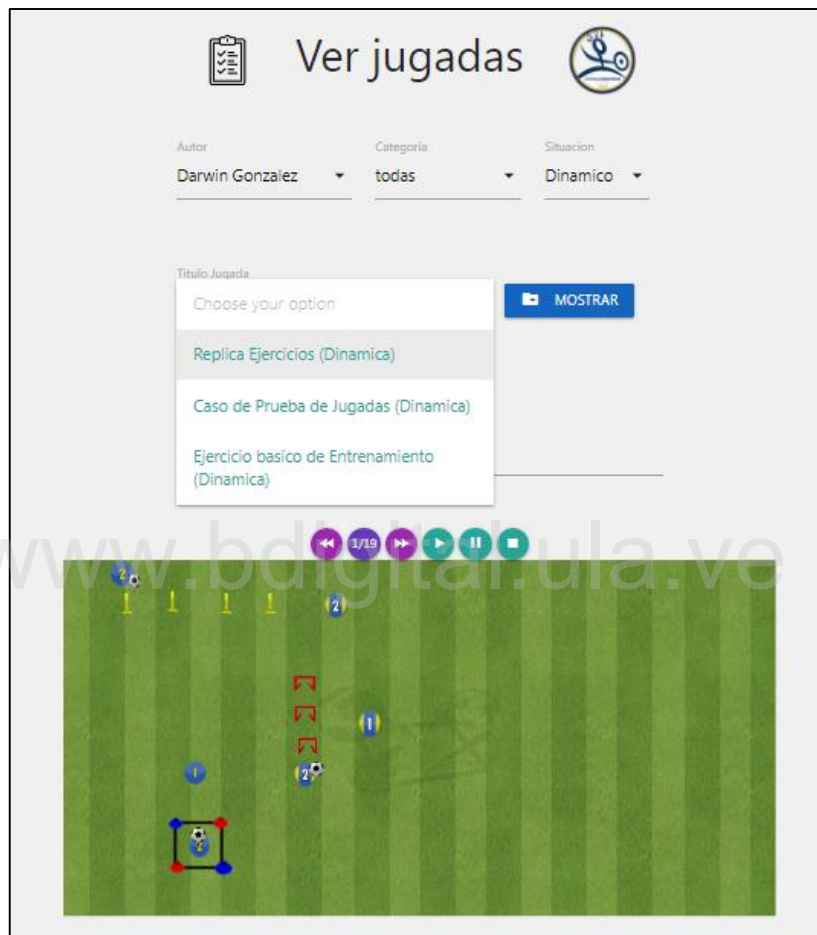
Campo Horizontal

Figura 4.8. Segunda parte Implementación Funcionalidad 1



4.3.2 Implementación Funcionalidad 2

Figura 4.9. Implementación Funcionalidad 2.



4.3.3 Implementación Funcionalidad 3

Figura 4.10. Primera parte Implementación Funcionalidad 3.



 **Editar Jugada** 

Seleccion Jugada para editar

Autor: Darwin Gonzalez ▼ Categoria: todas ▼ Situacion: Dinamico ▼

Titulo Jugada: Replica Ejercicios (Dinamica) ▼ **MOSTRAR**

Jugada

Titulo Nueva Jugada: Replica Ejercicios

Decripcion Jugada: Nuevos Movimientos

Categoria Jugada: Defensa ▼ Situacion: Dinamica ▼

Figura 4.11. Segunda parte Implementación Funcionalidad 3.



4.3.4 Implementación Funcionalidad 4

Figura 4.12. Implementación Funcionalidad 4.

The screenshot shows a web application interface with a dark gray header and footer. The header contains the text "Elige un Campo o Ambiente". The main content area is titled "Configuracion Elemento" and features a user profile section for "Carlos Ochoa". Below this, there is a row of buttons representing different positions: DEFENSA, LATERAL, LIBERO, MEDIO DEFENSIVO, INTERIOR IZQ, INTERIOR DER, EXTREMO, and CENTRO DELANTERO. The "SEGUNDO DELANTERO" button is highlighted. Underneath, the "Plantilla - Equipo" section lists four players: JOSE ANDRES PEREZ, CARLOS OCHOA (highlighted in blue), MANUEL PEREZ, and ROBERT DUARTE. At the bottom right, there are "CANCELAR" and "ACEPTAR" buttons. The footer contains a navigation bar with "FIGURAS", "EQUIPACION", and "JUGADORES" buttons. A large watermark "www.bdigital.ula.ve" is visible across the bottom of the interface.

4.3.5 Modelo de Datos

El modelo de datos que se utilizó para guardar la información de las *Jugadas* estuvo influenciado por la tecnología elegida para la base de datos, como se utilizó *MongoDB* los datos tuvieron la siguiente representación en forma de *collection* de esta tecnología.

Figura 4.13. Modelo de datos de una Jugada.

```

Titulo: String,           // Titulo Jugada
Descripcion: String,      // Descripcion Jugada
CategoriaJugada: String,  // Categoria Jugada
SituacionJugada:String,   // Situacion Jugada
CampoJugada: String,      // Ambiente trabajo
Movimientos: [Object],    // Secuencia Movimientos
AsociacionImagenes: Object, // Items
DescripcionMovimientos: Object, // Descripcion Secuencia M.
IdUsuario:String,        // Id Usuario Autor de Jugada

```

4.5 Pruebas

4.5.1 Pruebas de Aceptación

Durante esta fase se desarrollan las pruebas de aceptación de las historias de usuario que se elaboraron. En el cuadro 4.21 se definen de forma general todas las pruebas de aceptación y en las tablas 4.22- 4.25 se describe de manera detallada cada una de ellas.

Cuadro 4.21. Pruebas de Aceptación

Numero de Prueba	Numero de Historia Usuario	Nombre caso de Prueba
1	1	Crear y guardar una nueva Jugada
2	2	Seleccionar Jugada para visualizarla
3	3	Seleccionar Jugada para editar
4	3	Seleccionar Jugada para eliminar

Cuadro 4.22. Prueba de Aceptación 1

Pruebas de Aceptación	
Número: 1	Historia de Usuario: 1
Nombre caso de Prueba: Crear y guardar una nueva Jugada	
Condiciones de Ejecución: Se parte de la premisa que existen usuarios.	
Pasos de Ejecución: El usuario entra en la funcionalidad 1 y realiza las siguientes acciones: • Llena los campos de título y descripción de la Jugada. • Selecciona los valores situación y categoría que tendrá la Jugada. • Selecciona un ambiente de trabajo • Agrega y elimina ítems dentro del ambiente de trabajo seleccionado. • Personaliza ítems a su criterio. • Crea secuencia de movimientos con los ítems del ambiente de trabajo. • Guarda la Jugada en base de datos.	
Resultado Esperado: Se almacena en base de datos la información general de la Jugada elaborada al igual que la secuencia de movimientos de los ítems.	
Evaluación de la Prueba: La prueba se realizó satisfactoriamente.	

Figura 4.14. Primera parte prueba de aceptación 1.

The screenshot shows a web form titled "Crear Jugada" (Create Play). It includes a clipboard icon with a play symbol and a circular logo with a soccer ball. The form has several sections:

- Título Nueva Jugada** (New Play Title): A text input field containing "Caso de Prueba de Jugadas".
- Descripción Jugada** (Play Description): A text area containing "Esta es una jugada de prueba donde se verifica que todas las funcionalidades del software cumplen con lo planificado".
- Categoría Jugada** (Play Category): A dropdown menu set to "Defensa".
- Situación** (Situation): A dropdown menu set to "Dinamica".
- Elige un Campo o Ambiente** (Choose a Field or Environment): A section with five soccer field icons. The second icon, labeled "Campo Horizontal", is selected.

A large, semi-transparent watermark "www.bdigital.ula.ve" is visible across the bottom half of the form.

Figura 4.15. Segunda parte prueba de aceptación 1.



Figura 4.16. Tercera parte prueba de aceptación 1.

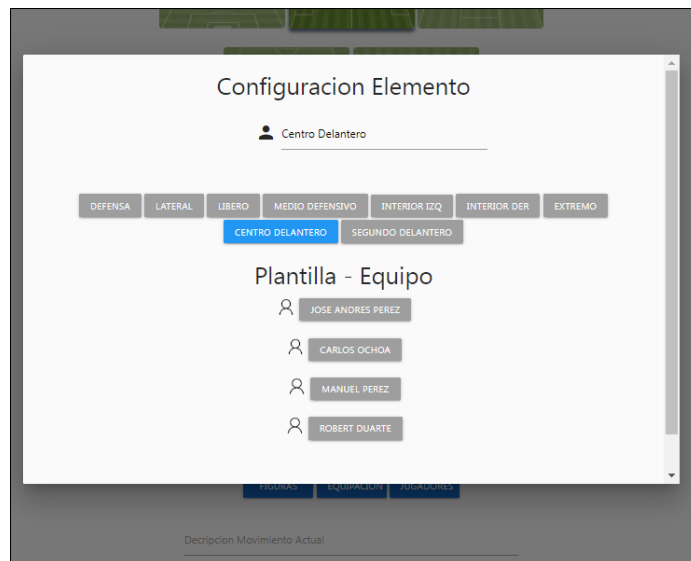


Figura 4.17. Cuarta parte Prueba de aceptación 1.



Cuadro 4.23. Prueba de Aceptación 2

Pruebas de Aceptación	
Número: 2	Historia de Usuario: 2
Nombre caso de Prueba: Seleccionar Jugada para visualizarla	
Condiciones de Ejecución: Se parte de la premisa que existe una jugada creada de un Usuario.	
Pasos de Ejecución: El usuario entra en la funcionalidad 2 y realiza las siguientes acciones: • Selecciona a un Usuario de una lista de usuarios que tienen Jugadas creadas. • Selecciona una Jugada de una lista de Jugadas del usuario seleccionado. • Visualiza el ambiente de trabajo con los botones para generar la secuencia de movimientos.	
Resultado Esperado: Se visualiza la jugada seleccionada y se puede reproducir la secuencia de movimientos.	
Evaluación de la Prueba: La prueba se realizó satisfactoriamente.	

Figura 4.18. Primera parte Prueba de aceptación 2.



Figura 4.19. Segunda parte Prueba de aceptación 2.



Cuadro 4.24. Prueba de Aceptación 3

Pruebas de Aceptación	
Número: 3	Historia de Usuario: 3
Nombre caso de Prueba: Seleccionar Jugada para editar	
Condiciones de Ejecución: Se parte de la premisa que existe una jugada creada de un Usuario.	
Pasos de Ejecución: El usuario entra en la funcionalidad 3 y realiza las siguientes acciones: • Selecciona una Jugada de su lista de Jugadas creadas. • Visualiza el ambiente de trabajo y modifica la información general y/o los ítems junto con la secuencia de movimientos. • Guarda en base de datos la Jugada modificada.	
Resultado Esperado: Se almacena en base de datos la información general y la secuencia de movimientos de los ítems de la Jugada modificada.	
Evaluación de la Prueba: La prueba se realizó satisfactoriamente.	

Figura 4.20. Primera parte Prueba de aceptación 3.

Figura 4.21. Segunda parte Prueba de aceptación 3.

Titulo Jugada

Caso de Prueba de Jugadas (Dinamica) MOSTRAR

Jugada

Titulo Nueva Jugada

Caso de Prueba de Jugadas (prueba Editado)

Descripcion Jugada

Esta es una jugada de prueba que fue editada donde se verifica que todas las funcionalidades del software cumplen con lo planificado

Categoria Jugada: Defensa

Situacion: Dinamica

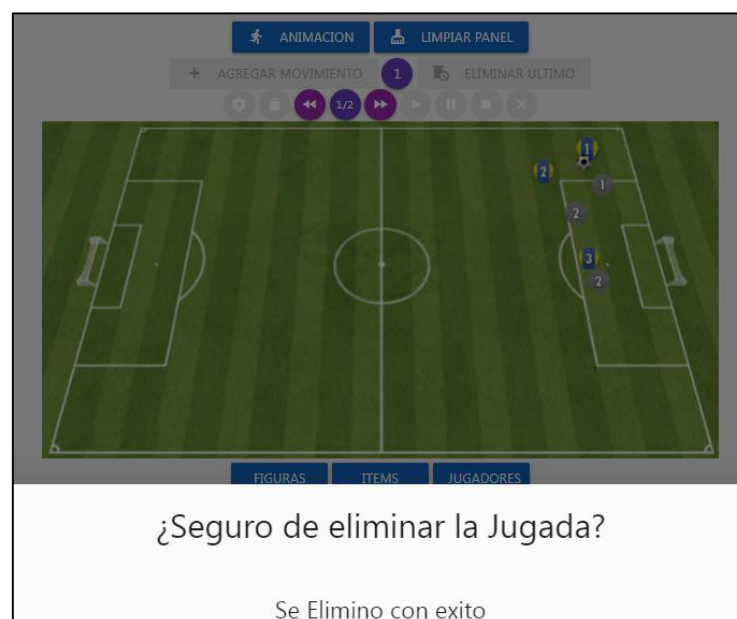
Figura 4.22. Tercera parte Prueba de aceptación 3.



Cuadro 4.25. Prueba de Aceptación 4.

Pruebas de Aceptación	
Número: 4	Historia de Usuario: 3
Nombre caso de Prueba: Seleccionar Jugada para eliminar	
Condiciones de Ejecución: Se parte de la premisa que existe una jugada creada de un Usuario.	
Pasos de Ejecución: El usuario entra en la funcionalidad 3 y realiza las siguientes acciones: • Selecciona una Jugada de su lista de Jugadas creadas. • Visualiza el ambiente de trabajo y modifica la información general y/o los ítems junto con la secuencia de movimientos. • Elimina la Jugada seleccionada.	
Resultado Esperado: Se elimina de la base de datos la Jugada seleccionada.	
Evaluación de la Prueba: La prueba se realizó satisfactoriamente.	

Figura 4.23. Prueba de aceptación 4.



4.5.2 Comparativa entre Softwares

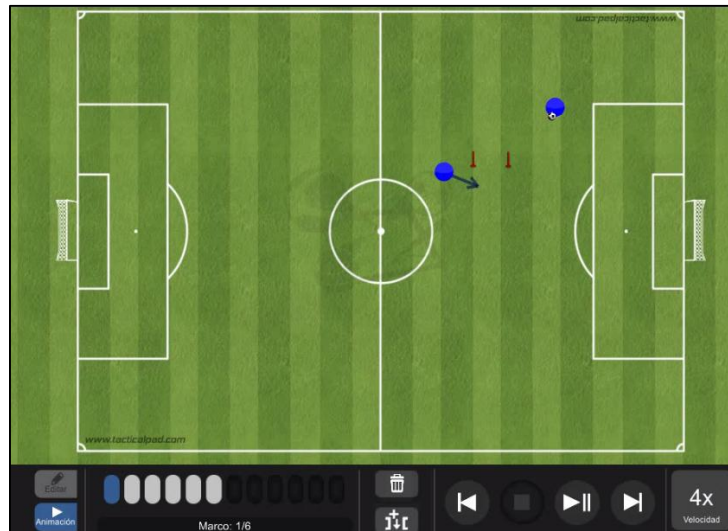
En estas pruebas se replicó una misma Jugada tanto en los softwares profesionales *TacticalPad* y *TheCoachPlatform* como en el software de análisis e implementación táctica que se desarrolló en este capítulo, con la finalidad de apreciar el alcance y las distancias que pueden existir entre el software elaborado y otros softwares de estas características que ya están en el mercado.

Primero se elaboró un ejercicio básico de entrenamiento que consta de una secuencia de 6 movimientos o frames/marcos. El software desarrollado ejecuto la secuencia de movimientos a la misma capacidad que los softwares profesionales, no se apreció grandes diferencias entre los 3 softwares utilizados.

Figura 4.24. *Ejercicio de enteramiento utilizando el software desarrollado*



Figura 4.25. Ejercicio de enteramiento utilizando el software TactitalPad



Nota. Elaborado en el software *TactitalPad*

Figura 4.26. Ejercicio de enteramiento utilizando el software TheCoachPlatform



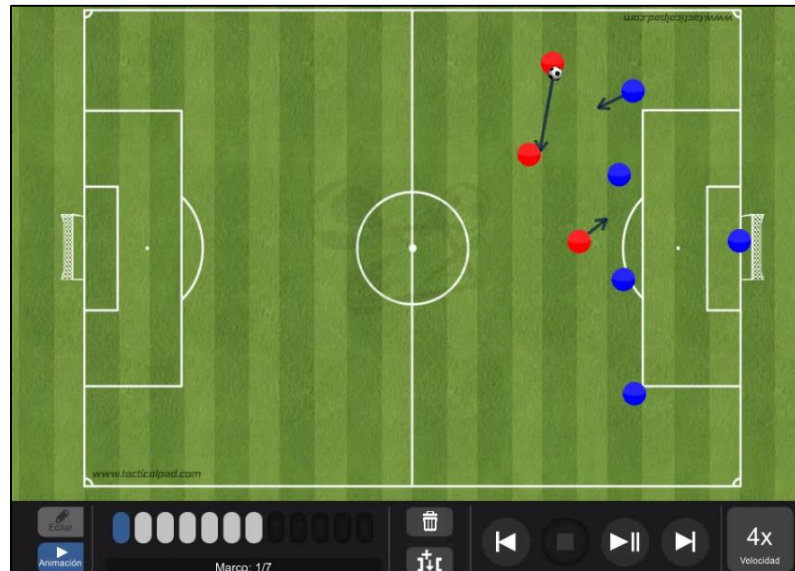
Nota. Elaborado en el software *TheCoachPlatform*

Por último, se elaboró un ejercicio táctico de ataque de una secuencia de 6 movimientos o frames/marcos. Se pudo apreciar el mismo resultado donde software desarrollado ejecuta la secuencia de movimientos a la misma capacidad que los otros softwares profesionales, sin apreciar grandes diferencias entre ellos.

Figura 4.27. Ejercicio táctico utilizando el software desarrollado

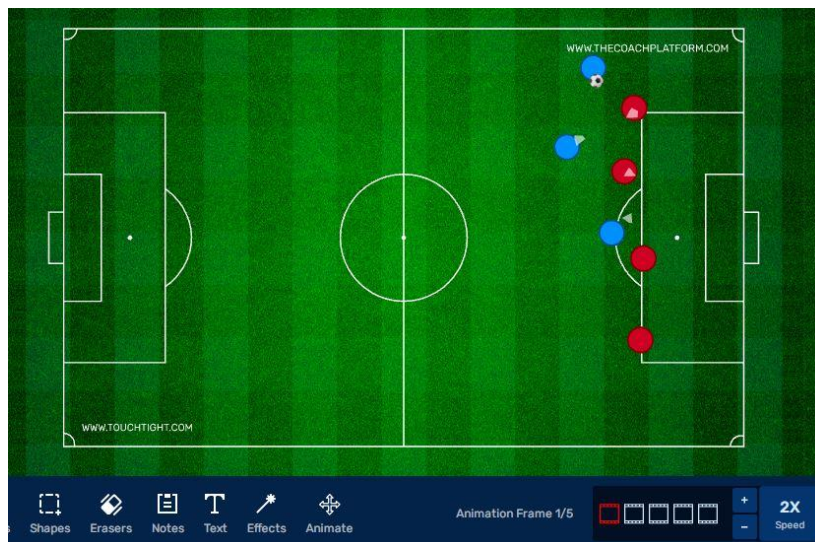


Figura 4.28. Ejercicio táctico utilizando el software TactitalPad



Nota. Elaborado en el software *TactitalPad*

Figura 4.29. Ejercicio táctico utilizando el software TheCoachPlatform



Nota. Elaborado en el software *TheCoachPlatform*

Capítulo 5

Conclusiones y Recomendaciones

La investigación tuvo como propósito desarrollar un software para el análisis e implementación táctica para ser presentado a los miembros del club de futbol Academia Emeritense FC, mediante el uso de la metodología ágil Extreme Programming se logró diseñar e implementar un software que cumple con las necesidades de esta entidad deportiva tanto para planificación como elaboración de gran parte de los aspectos tácticos y de entrenamiento que manejan en la Academia deportiva. Este software se elaboró utilizando las tecnologías HTML, CSS y JavaScript ya que estas son de simple integración con otros proyectos.

El software elaborado cuenta con tres funcionalidades básicas las cuales permiten crear un ambiente de trabajo y elaborar una secuencia de movimientos deseada, guardar esa secuencia y poder recrearla cuando sea necesario bien sea para su visualización o para su

edición. Estas funcionalidades presentan un gran nivel de desarrollo donde incluso pueden compararse con funcionalidades similares de softwares profesionales de este ámbito.

Como recomendaciones para este trabajo, se proponen las siguientes:

- Realizar la integración de este software con la plataforma deportiva de la Academia Emeritense FC una vez esta plataforma este completamente desarrollada.
- Elaborar imágenes propias de los fondos para los ambientes de trabajo y no utilizar imágenes de un tercero.
- Incorporar más funciones esenciales en el ambiente de trabajo como aumentar-disminuir la velocidad de reproducción de la animación o maximizar-minimizar el espacio del ambiente de trabajo para mayor comodidad visual para el usuario.
- Vincular más información del usuario sobre los equipos o jugadores que estén a su cargo para personalizar las funcionalidades.

Bibliografía

Barbero, Á. J., Vera, J. G., y Castagna, C. (2006). *Cuantificación de la Carga en Fútbol: Análisis de un Juego en Espacio Reducido*. Pensar en movimiento.

Bautista, Q. (2012). *Programación Extrema XP*. Unión Bolivariana.

Chiluisa, A., y Loarte, B. (2014). *Desarrollo e Implantación del Sistema de Control de Inventarios y Gestión de Laboratorios para la Facultad de Ciencias de la Escuela Politécnica Nacional*. Quito.

Dietrich, M., Klaus, C., y Klaus L. (2001). *Manual de metodología del entrenamiento deportivo*. Paidotribo.

Espar, F. (2004). *Entrenamiento de la táctica. Apuntes del Máster de Alto Rendimiento en Deportes Colectivos*. INEFC-FCB-Byo-medic.

Flanagan, D. (1998). *JavaScript: The Definitive Guide, Third Edition*, O'Reilly & Associates.

- Foster, D., y García, K. (2014). *Jornada de formación permanente sobre la metodología del entrenamiento deportivo aplicada al béisbol para los entrenadores de la escuela de béisbol menor 'los chamos' ubicada en la comunidad los próceres del municipio Naguanagua-edo.Carabobo*. Universidad de Carabobo, Venezuela.
- Geambasu, C., Jianu, I., Jianu, I., y Gavrilă, A. (2011). *Influence Factors for the choice of a Software Development Methodology*. Academia de estudios económicos de Bucarest.
- Izquierdo, M., y González, J. (2006). *Tecnologías y aplicaciones en la evaluación de la fuerza*. Junta de Andalucía.
- Johnson R. E., y Foote B. (1988). *Designing Reusable Classes*. Journal of Object-Oriented Programming.
- Meléndez V., Sinthya M., María E., y Pérez R., y Neldin N. (2016). *Sistema web de evaluación al desempeño docente unanmanagua, empleando la metodología ágil programación extrema*. Universidad Nacional Autónoma de Nicaragua.
- Platonov, V. (1993) *La preparación física. Colección Deporte y Entrenamiento*. Paidotribo.
- Pacheco R., y Márquez, W. (2010). *Capacitación a los Entrenadores de la Escuela de Béisbol Menor del Municipio Nirgua, a través de un Programa Teórico Practico Para Mejorar el Desarrollo de las Estrategias Metodológicas*. Universidad de Carabobo, Venezuela.
- Rishe, N. (1992). *Database Design: The Semantic Modeling Approach*. McGraw-Hill, Inc.

Rouse, M. (2 de septiembre de 2022). *Base de datos relacional*.

<https://www.computerweekly.com/es/definicion/Base-de-datos-relacional>

Rodríguez, P. L., y Moreno, J. A. (1996). *La iniciación al voleibol como medio para una formación integral en Educación Física*. Gráficas Yuliá.

Tejada, Z. (10 de Septiembre de 2022). *Datos no relacionales y NoSQL*.

<https://learn.microsoft.com/es-es/azure/architecture/data-guide/big-data/non-relational-data>

Ulloa, D. (2014). *Estudio de metodologías para estandarizar el desarrollo de software en el departamento de informática en la pastoral social caritas de la diócesis de Ambato*.

Universidad técnica de Ambato, Ecuador.

Weineck, J (2005). *Entrenamiento total*. Paidotribo.

Zintl, F. (1991). *Entrenamiento de la resistencia*. Barcelona: Martínez Roca.