



Universidad de las Ciencias
Informáticas

Sistema para el análisis estadístico de información para
estudiantes y profesores especialistas de futsal del
departamento de deportes de la Universidad de las
Ciencias Informáticas

SISFUT

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

Autor: Yalbert González Antúnez

Tutores: MsC. Gregorio Morales González

Ing. Gelson Rafael Saurín Ojeda

Co-Tutor: Ing. Yalena Aileth Venega Cañizares

La Habana, 2016
Año 58 de la Revolución



"Cuanto más difícil es la victoria, mayor es la felicidad de ganar"
Pelé

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste se firma la presente a los ____ días del mes de _____ del año _____.

Yalbert González Antúnez

Autor

MsC. Gregorio Morales
González

Tutor

Ing. Gelson Rafael Saurín Ojeda

Tutor

Ing. Yalena Aileth Venega

Cañizares

Co-Tutor

DATOS DE CONTACTO

Tutor: MsC. Gregorio Morales González

Edad:

Ciudadanía: cubana.

Institución: Universidad de las Ciencias Informáticas (UCI).

Título: Ingeniero en Ciencias Informáticas.

Categoría Docente: Instructor.

E-mail: alina@uci.cu

Graduado de la UCI. Posee 4 años de experiencia como Analista de Sistema en proyectos de software de Informática Industrial y Realidad Virtual. Analista principal del proyecto DISEM (Diseño y Simulación de Estructuras Mecánicas).

Tutor: Ing. Gelson Rafael Saurín Ojeda.

Edad: 31

Ciudadanía: cubana.

Institución: Universidad de las Ciencias Informáticas (UCI).

Título: Ingeniero en Ciencias Informáticas.

Categoría Docente: Profesor.

E-mail: grsaurin@uci.cu.

Graduado en ingeniero en ciencias informáticas, actualmente se encuentra de Profesor de Sistema Operativo y es jefe del Departamento de Construcción de Aplicaciones en el centro Entornos Interactivos 3D.

Co – Tutor: Ing. Yalena Aileth Venega Cañizares

Edad: 24

Ciudadanía: cubana.

Institución: Universidad de las Ciencias Informáticas (UCI).

Título: Ingeniero en Ciencias Informáticas.

E-mail: yalena@uci.cu.

Graduada de Ingeniero en Ciencias Informáticas (2015). Ha participado en eventos nacionales e internacionales y ha publicado varios artículos relacionados con los videojuegos serios. Actualmente se encuentra trabajando en el centro VERTEX de la Facultad 5, como analista de videojuegos y administradora de calidad en el proyecto de Realidad Aumentada.

DEDICATORIA

A mis padres, mi tía, mi abuela, mi tío, mi abuelo, alfre y a todas las personas que me quieren y siempre confiaron en mí brindándome su apoyo para que viera realizado mi sueño, quiero regalarles este momento y honrarlos por tanto amor y dedicación. Los quiero mucho.

AGRADECIMIENTOS

Después de muchos años de estudio y esfuerzo, hoy me gradúo de Ingeniero en Ciencias Informáticas. Para hacer realidad este sueño muchas personas me ayudaron, apoyaron y depositaron su confianza en mí. Hoy tengo la oportunidad de hacerles saber cuan agradecido estoy.

A mi madre, la persona que más quiero y admiro, gracias por quererme y apoyarme en todas mis decisiones, gracias por la dedicación y el cariño que siempre me has dado.

A mi abuela, por estar siempre pendiente de mí en todo momento, por estar siempre a mi lado cuando lo necesite.

A mi tía, por tu cariño y tu apoyo incondicional.

A alfre, por estar presente todos estos años y tratarme como un hijo.

A mi tío, por ser mi amigo, mi hermano, mi padre, siempre ha sido un ejemplo a seguir.

A mi abuelo, por su amor y cariño.

A Claudia, por ser lo más lindo que me ha pasado en la UCI, por soportarme y quererme, por apoyarme y darme fuerzas para seguir adelante.

Por otra parte, quería agradecerle al tribunal, a mis tutores, Yalena, Gelson, a Livan, Gregorio y Pochet por todo su esfuerzo y paciencia, ustedes son una de las piezas más importantes en esta tesis y quiero que sepan que son mis amigos por encima de todo.

También a las personas que forman el día a día en mi vida junto a mi familia, a mi piquete tanto los que están fuera de la escuela como los de aquí, Hayron, Nino, Handito, Alexis, Oscar, Alik, Romell, Silvio, Boteli, Leke, Mackay, Camilo, Ramón, Luiso, Ariel, Reimer, Luis Hansel y Hanoi los cuales hemos vivido grandes momentos juntos, nos hemos apoyado los unos a los otros y más que mis amigos los considero a todos mis hermanos y por último y no menos importante a la familia del fútbol en la escuela.

En fin, a todos aquellos que de una forma u otra han formado parte de mi vida y me han apoyado gracias a todos.

RESUMEN

El departamento de deportes de la Universidad de las Ciencias Informáticas (UCI) desea guardar la información referente a la práctica del fútbol en la universidad, de forma que esta se encuentre disponible para los profesores de educación física y especialistas encargados de la promoción de este deporte y que de esta forma puedan tomar mejores decisiones en cuanto a las competencias y el entrenamiento en las que participan atletas que frecuentemente se encuentran bajo condiciones de incertidumbre. En el presente trabajo se propone como objetivo el desarrollo de un sistema de gestión de la información para sus análisis y la toma de decisiones por estudiantes y profesores especialistas de fútbol.

Para llevar a cabo el desarrollo del sistema SISFUT se siguieron los pasos que propone la metodología XP en sus fases de trabajo. Se emplearon herramientas libres y de código abierto, cuya selección fue el resultado de un estudio comparativo entre las tendencias y tecnologías actuales. El desarrollo del trabajo proporcionó una herramienta que permitirá gestionar toda la información de las competencias y entrenamientos de fútbol, apoyando a los especialistas para que realicen una toma de decisiones más acertada en cuanto a la lógica interna de este deporte.

Palabras claves: competencias, fútbol, promoción.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN	4
1.1. INTRODUCCIÓN.....	4
1.2. CONCEPTOS RELACIONADOS CON EL DOMINIO	4
1.2.1. <i>Estadística</i>	4
1.2.2. <i>Sistema de información</i>	5
1.2.3. <i>Sistema de información estadístico (SIE)</i>	5
1.2.4. <i>Toma de decisiones</i>	5
1.2.5. <i>Incertidumbre</i>	7
1.2.6. <i>Razonamiento basado en casos</i>	8
1.2.7. <i>Sistemas basados en casos</i>	9
1.2.8. <i>Futsal</i>	10
1.3. ANÁLISIS DE SOLUCIONES EXISTENTES	10
1.3.1. <i>Handball Match Analysis: Computerized Notation System (HMACNS)</i>	11
1.3.2. <i>FutStats – HD</i>	12
1.3.3. <i>InnovaFUTSAL</i>	12
1.4. METODOLOGÍA DE DESARROLLO DE SOFTWARE	13
1.4.1. <i>Proceso Unificado Ágil</i>	13
1.4.2. <i>Programación Extrema</i>	14
1.4.3. <i>Scrum</i>	14
1.4.4. <i>Selección de la metodología de desarrollo</i>	15
1.5. ANÁLISIS DE ENTORNOS DE DESARROLLO INTEGRADO.....	15
1.6. HERRAMIENTAS Y TECNOLOGÍAS PARA EL DESARROLLO.....	17
1.7. CONCLUSIONES.....	20
CAPÍTULO 2: SOLUCIÓN PROPUESTA.....	21
2.1. SISFUT.....	21
<i>Roles del sistema</i>	21
<i>Variables analizadas para la toma de decisiones</i>	22
2.2. FASE I: EXPLORACIÓN.....	24
2.2.1. <i>Historias de Usuario</i>	24
2.3. FASE II: PLANIFICACIÓN	30
2.3.1. <i>Lista de reservas del producto</i>	30
2.3.2. <i>Desarrollo del Plan de duración de las Iteraciones</i>	34
2.3.3. <i>Plan de Entregas</i>	35
2.4. FASE III: ITERACIONES	35
2.4.1. <i>Tarjetas CRC</i>	35
2.5. ARQUITECTURA DE SOFTWARE	37
2.5.1. <i>Patrón Modelo Vista Controlador (MVC)</i>	37
2.6. PATRONES DE DISEÑO	39
2.6.1. <i>Patrones GRASP</i>	39
2.7. MODELO DE DATOS	41
2.8. CONCLUSIONES DEL CAPÍTULO.....	41

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN	42
3.1. FASE IV: IMPLEMENTACIÓN.....	42
3.1.1. <i>Tareas de Ingeniería</i>	42
3.2. ESTILOS DE PROGRAMACIÓN.....	45
3.2.1. <i>Definiciones de clases</i>	45
3.2.2. <i>Definición de métodos</i>	45
3.2.3. <i>Asignaciones a variables</i>	46
3.2.4. <i>Estructuras de control</i>	46
3.2.5. <i>Llamada a funciones</i>	47
3.3. FASE V: PRUEBAS.....	47
3.3.1. <i>Pruebas de aceptación</i>	48
3.4. RESULTADOS DE LAS PRUEBAS.....	51
3.5. CONCLUSIONES DEL CAPÍTULO.....	52
RECOMENDACIONES.....	54
REFERENCIAS BIBLIOGRÁFICAS.....	56
CAPÍTULO 4. ANEXOS.....	59
ANEXO I: HISTORIAS DE USUARIOS.	59
ANEXO II: TARJETAS CRC.....	71
ANEXO III: TAREAS DE INGENIERÍA.....	75
ANEXO IV: MODELO DE DATOS	79

ÍNDICE DE FIGURAS

FIGURA 1: <i>FUNCIONAMIENTO DE UN SBC</i>	9
FIGURA 2: <i>REPRESENTACIÓN DEL FUNCIONAMIENTO DEL PATRÓN MVC</i>	38
FIGURA 3: <i>DIAGRAMA DE CLASES QUE REPRESENTA EL PATRÓN CREADOR</i>	40
FIGURA 4: <i>DIAGRAMA DE CLASES QUE REPRESENTA EL PATRÓN CONTROLADOR</i>	40
FIGURA 5: <i>DEFINICIÓN DE LAS CLASES</i>	45
FIGURA 6: <i>DEFINICIÓN DE MÉTODOS</i>	46
FIGURA 7: <i>ASIGNACIÓN A VARIABLES</i>	46
FIGURA 8: <i>ESTRUCTURAS DE CONTROL</i>	47
FIGURA 9: <i>LLAMADA A FUNCIONES</i>	47
FIGURA 10: <i>RESULTADOS DE LAS PRUEBAS DE ACEPTACIÓN</i>	52
FIGURA 12: <i>MODELO DE DATOS</i>	79

ÍNDICE DE TABLAS

TABLA 1: ROLES DEL SISTEMA.....	22
TABLA 2: HU1 ADICIONAR TORNEO	25
TABLA 3: HU11 GENERAR PRINCIPALES LÍDERES.....	26
TABLA 4: HU12 GENERAR LAS ESTADÍSTICAS FINALES DE CADA PARTIDO.....	27
TABLA 5: HU13 GENERAR LOS 5 MEJORES JUGADORES DE UN EQUIPO.....	28
TABLA 6: HU14 GENERAR EVENTO COMPETITIVO	29
TABLA 7: LISTA DE RESERVAS DEL PRODUCTO.....	30
TABLA 8: REQUISITOS NO FUNCIONALES.	33
TABLA 9: PLAN DE DURACIÓN DE LAS ITERACIONES	34
TABLA 10: PLAN DE ENTREGAS.....	35
TABLA 11: TARJETA CRC INSERTAR TORNEOS	36
TABLA 12: TARJETA CRC INSERTAR PARTIDO	36
TABLA 13: TARJETA CRC INSERTAR ARBITROS.....	37
TABLA 14: TAREA DE INGENIERÍA #1.	42
TABLA 15: TAREA DE INGENIERÍA #2	43
TABLA 16: TAREA DE INGENIERÍA #3	43
TABLA 17: TAREA DE INGENIERÍA #4	44
TABLA 18: TAREA DE INGENIERÍA #5	44
TABLA 19: TAREA DE INGENIERÍA #6	44
TABLA 20: CASO DE PRUEBA PARA ADICIONAR UN TORNEO.....	48
TABLA 21: CASO DE PRUEBA PARA ADICIONAR UNA EDICIÓN	49
TABLA 22: CASO DE PRUEBA PARA ADICIONAR UN EQUIPO.....	49
TABLA 23: CASO DE PRUEBA PARA GENERAR EVENTO COMPETITIVO	50
TABLA 24: CASO DE PRUEBA PARA GENERAR PRINCIPALES LÍDERES	51
TABLA 25: HU7 ADICIONAR EDICIÓN.	59
TABLA 26: HU5 ADICIONAR EQUIPO.	60
TABLA 27: HU29 ELIMINAR EQUIPO.....	62
TABLA 28: HU30 ELIMINAR EDICIÓN	62
TABLA 29: HU14 GENERAR EVENTO COMPETITIVO	63
TABLA 30: HU38 LISTAR EDICIONES	64
TABLA 31: HU39 LISTAR EQUIPOS.....	64
TABLA 32: HU40 LISTAR JORNADAS.	65
TABLA 33: HU41 LISTAR CALENDARIO.....	66
TABLA 34: HU42 LISTAR JUGADORES.....	66
TABLA 35: HU43 LISTAR CONTRATOS.	67
TABLA 36: HU44 LISTAR ÁRBITROS.	68
TABLA 37: HU44 LISTAR DIRECTORES TÉCNICOS.....	68
TABLA 38: HU45 LISTAR NOTICIAS.	69
TABLA 39: HU12 GENERAR PRINCIPALES LÍDERES.....	70
TABLA 40: TARJETA CRC INSERTAR JUGADORES.....	71
TABLA 41: TARJETA CRC INSERTAR CONTRATO JUGADOR.....	73
TABLA 42: TARJETA CRC INSERTAR DT.....	73
TABLA 43: TAREA DE INGENIERÍA # 7	75
TABLA 44: TAREA DE INGENIERÍA # 8	75

TABLA 45: TAREA DE INGENIERÍA # 9	76
TABLA 46: TAREA DE INGENIERÍA # 10	76
TABLA 47: TAREA DE INGENIERÍA # 11	77
TABLA 48: TAREA DE INGENIERÍA # 12	77
TABLA 49: TAREA DE INGENIERÍA # 13	78
TABLA 50: TAREA DE INGENIERÍA # 14	78

INTRODUCCIÓN

La información deportiva a nivel mundial se hace cada vez más exacta, llegando a los interesados de esta rama con clara visión y siendo capaz de arrojar datos que son interesantes para los lectores y analistas de diferentes modalidades deportivas. El futsal es un deporte que nace en 1930 y sin tener siquiera un siglo de creado, goza de gran aceptación mundial y nacional.

En las universidades cubanas se practican una amplia gama de deportes, las cuales ayudan a fomentar la cultura física para mantener y mejorar la condición física de los jóvenes universitarios.

En consideración a lo anteriormente expuesto la Universidad de las Ciencias Informáticas (UCI), brinda una alta atención a la práctica de diferentes deportes, llevando a cabo la realización de una serie de copas deportivas a lo largo de cada curso docente con el fin de promover la práctica sistemática de los deportes. Los eventos que más destacan son: inter-años por facultades, mundialitos de futsal, baloncesto, béisbol y fútbol *soccer* entre otros menos conocidos, todo promoviendo la práctica sistemática de los deportes. Sin embargo, todos estos eventos no son debidamente divulgados por lo que se ve afectado la puesta en conocimiento de la realización y comportamiento de los mismos. Por esta razón, los aficionados de los deportes quedan insatisfechos y la participación en las copas resulta ser pobre lo que afecta el objetivo del trabajo del departamento de deportes.

El ejemplo más fehaciente es el deporte futsal, el cual es considerado uno de los más practicados por la comunidad universitaria. Estadísticas de la tesis en opción al título de Máster en Actividad Física en la Comunidad el profesor Gregorio Morales González (González, 2009), recogen una serie de datos en los que se ve el impacto del público en estos eventos. También hace alusión a informaciones estadísticas de la lógica interna de este deporte, que pueden ser utilizadas por los estudiantes y especialistas para analizar el desempeño de los jugadores de futsal.

Es importante almacenar todas estas informaciones de forma que se encuentre disponible para los profesores de educación física y especialistas que tomen decisiones en cuanto a la práctica y entrenamiento del futsal, así como para los estudiantes que practican frecuentemente dicha disciplina, siendo necesario una herramienta informática que gestione toda la información que se genere de los eventos relacionados con este deporte, por todo lo anteriormente expuesto se tiene la siguiente **situación problemática**:

- Los especialistas de la rama del deporte futsal en la universidad presentan problemas de gestión de datos importantes en el comportamiento de la dinámica del juego.
- Los profesores y entrenadores de futsal no tienen como analizar los datos obtenidos de los eventos

competitivos, juegos y entrenamientos del deporte futsal para la toma de decisiones y elaboración de estrategias de entrenamiento.

- Los especialistas de futsal en la universidad carecen de un sitio informativo propio para divulgar materiales importantes, sean videos de entrenamientos, conferencias, etc.
- El departamento de deportes de la universidad, no tiene como informar a la comunidad universitaria de toda información necesaria del deporte futsal.

Considerando la situación descrita, se ha definido el siguiente **diseño teórico de la investigación**:

- **Problema de la investigación:** ¿Cómo gestionar la información referente al deporte futsal, de forma que facilite a los profesores especialistas de este deporte en la toma de decisiones?
- **Objeto de estudio:** Sistemas informáticos de análisis estadístico basado en toma de decisiones.
- **Objetivo general de la investigación:** Desarrollar una aplicación para la toma de decisiones basado en análisis estadísticos de los jugadores.
- **Campo de acción:** Sistema informático de análisis estadístico para la toma de decisiones para el deporte futsal.

Objetivos específicos:

1. Elaborar el Marco Teórico de la investigación a partir de un estudio del estado del arte de los sistemas informáticos para análisis estadístico basado en toma de decisiones.
2. Analizar y seleccionar la metodología, herramientas y tecnología a utilizar para el desarrollo del sistema de gestión SISFUT.
3. Realizar el análisis y diseño de la aplicación.
4. Diseñar la base de datos que soporte las funcionalidades del sistema de gestión SISFUT.
5. Validar la herramienta propuesta a través de los métodos definidos en la investigación.

En la investigación se destaca la utilización de los siguientes **métodos teóricos**:

- **Analítico-Sintético:** Se utilizó en la búsqueda y análisis de los elementos más importantes asociados a la elaboración de software relacionados con la gestión y análisis estadísticos del futsal según la bibliografía consultada en libros, páginas web y artículos, realizando una síntesis de los elementos más importantes de la misma.

- **Histórico-Lógico:** Permitió una mayor comprensión del estado y tendencias actuales de software relacionados con la gestión y análisis estadísticos del fútbol, así como las tecnologías existentes y utilizadas lo que permitió la selección de las más idóneas para el desarrollo del sistema.
- **Modelación:** Se utilizó para realizar una representación abstracta de los conceptos y artefactos que se generaron en el desarrollo de la investigación y la solución.

Los **métodos empíricos** utilizados para obtener información sobre el objeto de estudio fueron:

- **Consulta bibliográfica:** Permite la elaboración del marco teórico de la investigación fundamentada por la información consultada.
- **Observación científica:** Se tuvo en cuenta a la hora de valorar la propuesta más adecuada de las tecnologías y soluciones existentes, además de permitir la observación de diferentes sitios web para conocer su funcionamiento, organización y estructura de la información que contiene.
- **Entrevista:** Se empleó en encuentros con los integrantes de la dirección de deporte de la Universidad con el objetivo de entender mejor el proceso del negocio y definir las principales funcionalidades del sistema.

El presente trabajo de diploma está compuesto por tres capítulos que incluyen los procedimientos desarrollados en relación con el trabajo investigativo, así como la propuesta de solución y validación de la investigación.:

Capítulo 1: “Fundamentación teórica de la investigación”: En este capítulo se abarcará el estado de los sistemas de análisis y gestión de estadísticas del fútbol a nivel nacional e internacional, para posteriormente realizar un balance y estudio de un conjunto de herramientas, tecnologías y metodología a seleccionar, para el desarrollo de la propuesta de solución.

Capítulo 2. “Solución propuesta”: En este capítulo se expone la solución propuesta y características del sistema de selección de los sistemas de análisis y gestión de estadísticas del fútbol, describiéndose los requisitos funcionales y no funcionales, diagrama de modelo de datos, así como todo lo referente a la arquitectura del sistema y a los patrones de diseño utilizados.

Capítulo 3: “Validación de la solución”: En este capítulo se detalla la construcción y validación de la solución al problema planteado. Además, se realizan y describen las pruebas de software a la aplicación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

1.1. Introducción.

En este capítulo se definen los principales conceptos que serán empleados durante todo el trabajo y se presentan las bases teóricas fundamentales relacionadas a la Inteligencia Artificial. Se realiza además un estudio del estado del arte acerca de las principales tendencias de la misma, las herramientas y las técnicas utilizadas para su desarrollo. Se realiza un estudio de sistemas similares, factor importante para que el desarrollo de la propuesta a presentar obtenga una mayor calidad, facilidad de uso y acceso a la información cumpliendo con las necesidades requeridas.

1.2. Conceptos relacionados con el dominio

1.2.1. Estadística

La estadística es la ciencia que tiene como objetivo reunir la información cuantitativa concerniente a individuos, grupos y series de hechos; y deducir de ello, basado en el análisis de estos datos, significados precisos o previsiones para el futuro (Morán, 2009).

Aunque las definiciones dadas por diferentes autores son semejantes, tienen variaciones de uno a otro. Para José Enrique Chacón, la estadística se define como “la ciencia que tiene por objeto el estudio cuantitativo de colectivos” (Ruiz Muñoz, 2004).

Sin embargo, Harald Crámer establece que “el principal objeto de la teoría estadística consiste en la investigación de la posibilidad de obtener inferencias válidas a partir de los datos estadísticos, y en la construcción de métodos para realizar dichas inferencias” (Crámer, 1999).

Óscar Vázquez Mínguez define la estadística como “la ciencia que tiene por objeto aplicar leyes de la cantidad a los hechos sociales para medir su intensidad, deducir las leyes que lo rigen y hacer su predicción próxima” (Ruiz Muñoz, 2004).

- **Estadística inferencial:**

La estadística inferencial o inductiva plantea y resuelve el problema de establecer previsiones y conclusiones generales sobre una población a partir de los resultados obtenidos de una muestra. La inferencia estadística es la metodología tendente a hacer descripciones, predicciones,

comparaciones y generalizaciones de una población estadística a partir de la información contenida en una muestra (Borrego del Pino, 2008).

Resumiendo, la definición de estadística inferencial, la Dra. Josefa Marín Fernández limita su concepto a: “parte de la estadística que se ocupa de llegar a conclusiones (inferencias) acerca de las poblaciones a partir de los datos de las muestras extraídas” (Fernández, 2010-2011).

- **Estadística descriptiva:**

La estadística descriptiva es un conjunto de métodos cuyo objetivo es ordenar las observaciones, resumir la información disponible y obtener las medidas cuantitativas que describen sus características. La descripción es cuantitativa y pretende ser lo más sintética posible (Munar, 2002).

La Dra. Josefa Marín Fernández resume la estadística descriptiva como “la parte de la estadística que se ocupa de recoger, clasificar, representar y resumir los datos de las muestras” (Fernández, 2010-2011).

1.2.2. Sistema de información

Un sistema de información se conforma con el conocimiento adquirido sobre un hecho y con elementos de análisis, evaluación, comparación y control, que permiten la toma de decisiones sobre un conjunto de actividades del sistema de servicios donde éste opera y conduce al logro de la finalidad y objetivos del mismo. Un sistema de información debe brindar información veraz, oportuna, relevante, exacta, útil y periódica (Payrol Trillo, y otros, 2008).

1.2.3. Sistema de información estadístico (SIE)

Todo sistema de información he de tener un subsistema que se encargue de la recolección, flujo y procesamiento de datos, que permita generar información numérica oportuna, confiable y necesaria. Este subsistema es llamado Sistema Estadístico. Para que el Sistema Estadístico se convierta en SIE, debe además contener elementos de análisis, evaluación, comparación y control que permitan el mejor desempeño del proceso de toma de decisiones (Payrol Trillo, y otros, 2008). Al tener este aspecto en cuenta, se hace necesario comprender en qué consiste la toma de decisiones.

1.2.4. Toma de decisiones

La toma de decisiones es el proceso mediante el cual se realiza una elección entre las opciones o formas para resolver diferentes situaciones de la vida en diferentes contextos: a nivel laboral, familiar, sentimental, deportivo o empresarial utilizando metodologías cuantitativas que brinda la administración. La toma de decisiones consiste, en elegir una opción entre las disponibles, a los efectos de resolver un problema actual o potencial. Es decir, la toma de decisiones consiste en la selección adecuada entre dos o más alternativas (Bello Pérez, 2001).

El proceso de toma de decisiones utiliza como materia prima la información, esta es fundamental ya que sin ella no resultaría posible evaluar las opciones existentes o desarrollar nuevas opciones. A mayor y mejor información, mayor calidad en la definición del problema, en las respuestas de solución, en el análisis de las variantes y en la selección de la acción más convincente (Bello Pérez, y otros, 2002).

En las organizaciones o personas que se encuentran sometidas constantemente a la toma de decisiones la información adquiere un rol fundamental y por ello un valor inigualable. Para procesar los datos de la organización y transformarlos en información, es fundamental el uso de sistemas de información. Además de los sistemas de información, existen sistemas diseñados especialmente para ayudar a transitar en el proceso de toma de decisiones, que se conocen como sistema de soporte a decisiones o sistemas de apoyo a la decisión. Las decisiones se pueden dividir en grandes grupos que se describen a continuación.

- **Tipos de decisiones:**

Las decisiones se pueden clasificar teniendo en cuenta diferentes aspectos como es la frecuencia con las que se representan, así como las circunstancias que afrontan estas decisiones, los principales tipos de decisiones son:

- **Decisiones programadas:** son las que se refieren a dar soluciones a problemas de rutinas determinadas por reglas, procedimientos o hábitos.
- **Las decisiones no programadas:** también denominadas no estructuradas, son aquellas que se toman ante problemas o situaciones que se presentan con poca frecuencia, o aquellas que necesitan un modelo o proceso específico de solución. Si un problema no se ha presentado con la frecuencia suficiente como para que lo cubra una política, o si se considera muy importante que merece un trato especial, este debe ser manejado como una decisión no programada.

- **Situaciones o contexto de decisión:**

Las situaciones o ambientes en los cuales se toman las decisiones se pueden clasificar según el conocimiento y control que se tenga sobre las variables que intervienen o influyen en el problema, debido a que la decisión final o solución va a estar condicionada por dichas variables.

Los principales factores que intervienen en las situaciones de decisión son:

- **Ambiente de certeza (Certidumbre):** se tiene conocimiento total sobre el problema, las opciones de solución que se planteen van a causar siempre resultados conocidos e invariables. Al tomar la decisión solo se debe pensar en la opción que genera mayor beneficio. La información con la que se cuenta para solucionar el problema es incompleta, es decir, se conoce el problema, se conocen las posibles soluciones, pero no se conoce con certeza los resultados que pueden arrojar (Bello Pérez, 2001).
- **Ambiente de incertidumbre:** Se posee información deficiente para tomar la decisión, no se tiene ningún control sobre la situación, no se conoce cómo puede variar la interacción de las variables del problema, se pueden plantear diferentes opciones de solución, pero no se le puede asignar probabilidad a los resultados que arrojen.

La incertidumbre puede manifestarse de diversas formas y ser provocada por diferentes causas, pero en grandes rasgos se pueden conocer tres categorías de incertidumbre:

- Aquella provocada por información o conocimiento impreciso.
- La provocada por información incompleta.
- La provocada por conceptos o palabras que son inherentemente inexactos.

1.2.5. Incertidumbre

Por su importancia en la toma de decisiones, se hace necesario comprender mejor la utilización de la incertidumbre en la realización de análisis estadísticos. Existe un ambiente de incertidumbre cuando falta el conocimiento seguro y claro respecto del desenlace o consecuencias futuras de alguna acción o situación, lo que puede derivar en riesgo cuando se aprecia la perspectiva de una contingencia con posibilidad de generar pérdidas o la proximidad de un daño. La incertidumbre supone cuantificar hechos mediante

estimaciones para reducir riesgos futuros, y aunque su estimación sea difícil no justificará su falta de información (Bello Pérez, 2001).

El concepto de incertidumbre, como un atributo cuantificable, es relativamente nuevo en la historia de las mediciones, aunque los términos de error y análisis de error han sido muy usados como parte práctica de la ciencia de las mediciones o metrología. Cuando se han evaluado todas las componentes conocidas y supuestas de un error y se han aplicado las correcciones adecuadas, todavía queda como remanente una incertidumbre sobre la corrección del resultado establecido, esto es, la correspondencia que existe entre el resultado de la medición de la magnitud en cuestión y la duda que se genera sobre este valor (Bello Pérez, 2001). Por tanto, la incertidumbre cuantifica una idea de la calidad y confianza del resultado, es decir al reflejar en cuanto puede alejarse el resultado analítico del valor considerado verdadero.

La incertidumbre puede ser tratada por diferentes técnicas que se definen en la Inteligencia Artificial para este propósito, dentro de las que se encuentra el Razonamiento Basado en Casos (RBC), el mismo se describe a continuación para una mejor comprensión.

1.2.6. Razonamiento basado en casos

El RBC es el proceso de solucionar nuevos problemas basándose en las soluciones de problemas anteriores. Por ejemplo, un mecánico de automóviles que repara un motor porque recordó que otro auto presentaba los mismos síntomas está usando RBC. Un abogado que apela a precedentes legales para defender alguna causa está usando RBC. También un ingeniero cuando copia elementos de la naturaleza, está tratando a esta como una “base de datos de soluciones”. El RBC es una manera de razonar haciendo analogías. Se ha argumentado que este tipo de razonamiento no sólo es un método poderoso para el razonamiento de computadoras, sino que es usado por las personas para solucionar problemas cotidianos. Mientras que de forma más radical se ha sostenido que todo RBC porque está basado en la experiencia previa (Aamodt, y otros, 1994).

El RBC cuenta con las siguientes características:

- La búsqueda de la solución a un problema no se inicia a partir de los datos o el objetivo, por lo que el camino se acorta considerablemente.
- El razonamiento incluye un algoritmo para determinar la medida de semejanza entre dos objetos.
- Memoria indexada.

- Memoria dinámica.

1.2.7. Sistemas basados en casos

Para apoyar esta técnica de la Inteligencia Artificial, en la rama de la informática se han desarrollado diferentes sistemas que implementan este tipo de razonamiento, donde estas aplicaciones deben respetar los procedimientos que se realizan en el análisis que propone el RBC. Los Sistemas Basados en Casos (SBC) basan su funcionamiento en experiencias registradas, ya sea por el propio sistema o bien por la persona experta; y a partir de este conocimiento sobre realizar una asociación con estas experiencias para extraer una solución factible. Estos sistemas son los que más se asemejan al razonamiento natural que poseen los seres humanos, por ejemplo, los niños razonan según sus experiencias a partir del trato con el entorno que los rodea. Si nos ponemos a pensar en los niños pequeños, el tipo de razonamiento que utilizan es el basado en las experiencias que le proporciona su trato con el medio (Aamodt, y otros, 1994).

Para desarrollar un SBC se deben definir los siguientes módulos:

- Módulo recuperador de casos.
- Módulo de adaptación.
- Módulo de evaluación de soluciones.
- Módulo de almacenamiento.

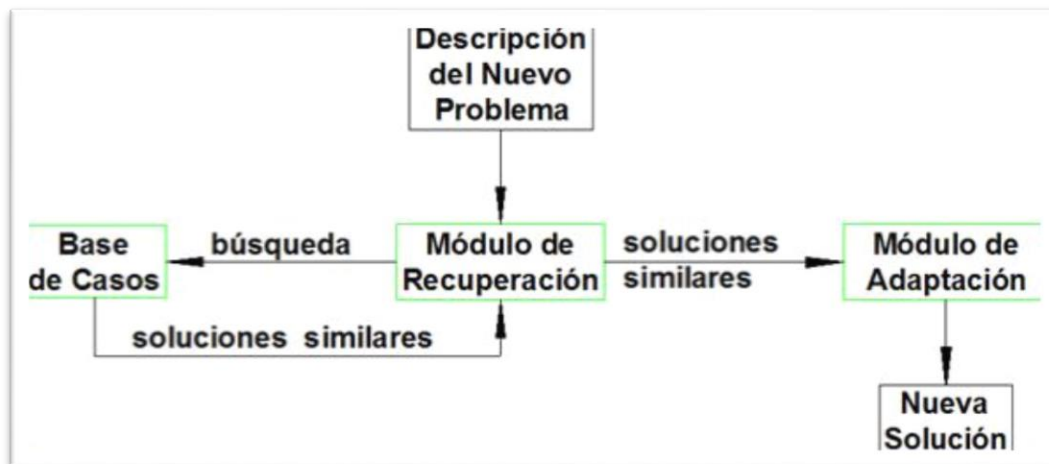


Figura 1: Funcionamiento de un SBC

1.2.8. Futsal

El fútbol sala nació en Uruguay en 1930. El entrenador de Fútbol tradicional creó el reglamento a partir del estudio de las reglas del waterpolo, baloncesto, balonmano y fútbol.

Término futsal es el término internacional usado para el juego. El juego era denominado en sus inicios como fútbol de salón (Paganelli, 2000).

Características de juego

En un partido de futsal hay 12 jugadores por cada equipo, 5 en el campo de juego y 7 en el banquillo. En función de la situación de juego (ataque o defensa), los jugadores ocupan posiciones diferentes en el campo (Paganelli, 2000).

Las diferentes posiciones en el juego son las siguientes:

- **Portero:** Encargado de proteger la portería y evitar que el balón traspase totalmente la línea de gol.
- **Cierre:** Jugador más atrasado en el equipo (excepto el portero) en el ataque.
- **Atacante derecho (Ala derecha):** Jugador que ataca por la banda derecha.
- **Atacante izquierda (Ala izquierda):** Jugador que ataca por la banda izquierda.
- **Pívot:** Jugador más adelantado y rápido del equipo. Es el jugador con más habilidades y control del balón.

1.3. Análisis de soluciones existentes

En el futsal al igual que en otros deportes, se utilizan las estadísticas como instrumento que posibilita al entrenador evaluar el comportamiento de sus jugadores en particular y del equipo en general, tanto en las competiciones como en la etapa de preparación. Los números que se obtienen en la recogida de información carecen de sentido si no hay un análisis e interpretación de los mismos para arribar a una conclusión que permita realizar correcciones y mejoras en el proceso de entrenamiento.

El desarrollo de la informática ha propiciado mejoras en el análisis estadístico y la aplicación de algunas de sus ventajas al ser utilizadas en diversos ámbitos y actividades humanas, donde el deporte no ha estado ajeno a este fenómeno. Las primeras aplicaciones de la informática en la esfera deportiva se originaron

durante la década de los 60 vinculados casi exclusivamente a la investigación en las ciencias aplicadas (Sharp, 1996), como consecuencias del creciente interés social por la actividad deportiva de rendimiento.

En un principio el empleo de medios informáticos se encontraba sólo al alcance de instituciones, laboratorios y grandes clubes profesionales, en gran medida debido al incipiente nivel de desarrollo tecnológico de la época. A partir de la aparición y distribución de los ordenadores personales a principios de los años 80, junto con la notable disminución de los costos de producción de los programas informáticos, comenzaron a desarrollarse numerosos estudios científicos encaminados a comprender y mejorar el rendimiento deportivo (Donnelly, 1987).

BrodieThornhill, en una publicación realizada en 1983, reconoce un elevado potencial de aplicación de la informática en el ámbito deportivo y anticipan futuros usos de los ordenadores como un nuevo elemento de equipamiento deportivo, también considera la capacidad de los software informáticos para gestionar la información generada por los entrenamientos y competiciones, además del análisis de los movimientos técnicos de los deportistas y para la valoración de las demandas físico – condicionales de una disciplina deportiva. (Brodie, y otros, 1983)

Después del año 2000 comenzó un auge de sistemas informáticos capaz de analizar todo tipo de datos deportivos, enfocados a varias modalidades destacando los dedicados a fútbol, baloncesto, balonmano, fútbol americano, béisbol y futsal. Con el objetivo de identificar posibles soluciones que den solución al problema de investigación, a continuación, se analizan algunos sistemas orientados a la gestión y análisis de la información del desempeño de los deportistas en diferentes modalidades:

1.3.1. *Handball Match Analysis: Computerized Notation System (HMACNS)*

Este programa está enfocado solamente para el balonmano, en ambos sexos. El almacenamiento de las acciones se realiza en un archivo de datos Access. Los informes son generados en un formato y pueden ser importados de otros paquetes de transformación de datos. Dos observadores son suficientes para registrar el partido. El programa realiza una relación de los ataques posicionales, los contraataques, y contraataques con mayor índice de eficiencia, así como la eficacia de la duración de los ataques del equipo en función de los sistemas de defensa que aplica el contrario, la eficacia de las acciones en los ataques posicionales en la zona 6 y 9 metros, la eficacia de los sistemas de defensa del equipo defensor y la eficacia de jugar en superioridad o inferioridad numérica. Permite acumular y sistematizar los datos estadísticos de

los partidos y separarlos según el equipo observado (Handball Match Analysis: Computerized Notation System, 2003).

1.3.2. *FutStats – HD*

FutStats – HD es una herramienta de análisis estadísticos de fútbol y futsal disponible solamente para algunos dispositivos como iPad, iPod o iPhone. La herramienta brinda la posibilidad de crear uno o varios equipos y luego crear los jugadores y asignarlos a sus correspondientes equipos. Cada jugador guardará sus estadísticas históricas, permitiendo que si un jugador pasa a otro equipo en el cual el entrenador utilice FutStats-HD, los datos pueden ser transferidos. Con el FutStats – HD se pueden guardar las acciones individuales y colectivas que realizan durante el partido o entrenamiento, mostrando luego las estadísticas y los análisis de eficiencia y rendimiento que están configurados en la herramienta (FutStats, 2013).

1.3.3. *InnovaFUTSAL*

Es una aplicación disponible para los sistemas operativos Windows y Mac OS. El análisis de datos y estadísticas está dirigido solamente al futsal, sin importar sexo o edad. Sus algoritmos de análisis están diseñados para actuar durante el juego, apoyándose en lo que ha sucedido durante el partido y los datos históricos de otros juegos y entrenamientos; lo que permite al entrenador tomar decisiones en tiempo real. El desarrollo de la aplicación está avalado por cuerpos técnicos del mundo de futsal y centrada en la máxima usabilidad (InnovaFutsal, 2015).

Después de haber analizado algunos sistemas que basan su negocio en la gestión y análisis de información generada por la actividad de los deportistas en los partidos que celebran, se obtuvieron evidencias del uso de las funcionalidades que se pretendían estudiar. Sin embargo, no es posible acceder a estos sistemas por tener un carácter propietario, por lo que sólo se puede comprobar de forma empírica cómo se gestionan las características deseadas. De acuerdo a esta situación, se decide utilizar mecanismos propios para la gestión de la información, así como definir las funciones de medición para evaluar las características de rendimiento a medir en los futbolistas.

Una de las tareas más importantes a tener en cuenta para el desarrollo de software, es realizar el trabajo guiado por un grupo de principios y procedimientos que garanticen un producto técnicamente probado y de

elevada calidad. Con el fin de aplicar los procedimientos adecuados para la construcción de la solución, se realiza un análisis de algunas metodologías concebidas con este propósito.

1.4. Metodología de desarrollo de software

Una metodología es el conjunto de fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de un sistema (Schenone, 2004). En ingeniería de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información (cms, 2008). El desarrollo de software es un proceso difícil y lleno de riesgos donde la metodología que se aplique para realizar lo mismo, define quién debe hacer qué, cuándo y cómo debe hacerlo. El fracaso en el desarrollo de un sistema está estrechamente relacionado con una mala selección de la metodología (Rodríguez-Perez, 2008), por lo que este proceso se debe realizar a partir de un análisis previo debe llevar un análisis previo y exhaustivo por parte del equipo de desarrollo. Las metodologías de desarrollo de software se clasifican en dos grandes grupos:

- **Pesadas/Tradicionales:** son las más tradicionales, se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar, y requiere una extensa documentación, ya que pretende prever todo de antemano. Este tipo de metodologías son más eficaces y necesarias cuanto mayor es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear, donde una gran organización es requerida (Pressman, 2003).
- **Ligeras/Ágiles:** orientadas a la interacción con el cliente y el desarrollo incremental del software, mostrando versiones parcialmente funcionales del producto mostrando en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios según se va desarrollando (Figuroa, y otros, 2008).

Teniendo en cuenta el tiempo con el que se cuenta para el desarrollo del presente trabajo y las condiciones del entorno, se restringe el análisis a metodologías ágiles para determinar la que mejor se adapte para desarrollar la solución del problema planteado.

1.4.1. Proceso Unificado Ágil

La metodología Proceso Unificado Ágil (AUP, por sus siglas en inglés) es una versión simplificada del Proceso Racional Unificado (RUP, por sus siglas en inglés) usando técnicas ágiles y conceptos que aún se mantienen válidos en esta última. El proceso unificado es un marco de desarrollo de *software* iterativo e

incremental, que a menudo es considerado como un proceso altamente ceremonioso debido a que especifica muchas actividades y artefactos involucrados en el desarrollo de un *software*. AUP puede ser adaptable y se preocupa especialmente de la gestión de riesgos, proponiendo que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y se aborden en etapas iniciales. Para ello, se crean y mantienen listas identificando los riesgos desde etapas iniciales del proyecto. Resulta especial el sentido que se presta al desarrollo de prototipos ejecutables durante la fase de elaboración del producto, donde se demuestre la validez de la arquitectura para los requisitos clave del producto y que determinan los riesgos técnicos (Cordero, J., 2010).

1.4.2. Programación Extrema

La Programación Extrema (XP, por sus siglas en inglés), es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito. En el desarrollo de software, promueve el trabajo en equipo y propicia un buen clima de trabajo. XP se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, estimula la comunicación fluida entre los participantes y aboga por la simplicidad en las soluciones implementadas, aunque demanda coraje para enfrentar los repentinos cambios que puedan surgir en el ciclo de vida del proyecto que se desarrolle. Esta metodología es recomendada de forma general para proyectos pequeños con requisitos imprecisos y cambiantes donde existe un alto riesgo técnico (Loaiza, 2010).

1.4.3. Scrum

“Scrum es un proceso de desarrollo de *software* iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de *software*. Aunque Scrum estaba enfocado a la gestión de procesos de desarrollo de *software*, puede ser utilizado en equipos de mantenimiento de *software* o en una aproximación de gestión de programas: Scrum de Scrums” (Schwaber, 1997).

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (llamados *requirements churn*) y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes (Sutherland, 2007).

1.4.4. Selección de la metodología de desarrollo

Se decidió utilizar XP debido a que se adapta en gran medida tanto al tipo de proyecto a desarrollar como a las condiciones de trabajo. A continuación, se exponen varias de las razones que llevaron al uso de esta metodología:

- El proyecto es pequeño.
- La metodología ágil más radical y popular. XP se centra en el ciclo de vida del desarrollo de software.
- El cliente forma parte del equipo de desarrollo. Mediante la aplicación de XP se puede lograr una retroalimentación mayor y lograr un producto que satisfaga sus necesidades.
- La gran ventaja de XP es su increíble capacidad de respuesta ante imprevistos, aunque por diseño es una metodología que no construye para el largo plazo y para la cual es difícil documentar.
- Consiste en diseñar, implementar y programar lo más rápido posible.
- Poca disponibilidad de personal. El sistema debe ser realizado por dos personas solamente, no siendo posible la existencia de muchos roles ni la especialización en un rol específico por parte de los miembros. Uno de los principios básicos de XP es la programación en equipos pequeños (2 a 12 personas) con pocos roles, pudiendo los miembros del equipo intercambiar responsabilidades en un momento determinado (GOMEZ, 2010).

Una vez seleccionado la forma de desarrollo que guiará el proceso ingenieril, es necesario identificar las herramientas y tecnologías que permitirán llevar a cabo esta tarea y por consiguiente se realiza un análisis de los diferentes entornos de desarrollo para elegir el que mejor se adapte a las necesidades del proyecto. Esta selección es vital para apoyar en la reducción del tiempo de implementación.

1.5. Análisis de Entornos de Desarrollo Integrado

Un Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse de forma exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios. Una decisión importante a la hora de desarrollar con PHP es la selección del IDE, ya que el entorno de desarrollo que se elija puede suponer una verdadera

diferencia en el tiempo de trabajo invertido. Debido a lo anterior se realiza un estudio sobre algunos de los IDE para PHP:

- **Zend Studio:** es el soporte en desarrollos y pruebas de PHP con el set más completo de herramientas para la creación de aplicaciones de alta fiabilidad como lo requiera una empresa. Asegura el desarrollo de software mediante la combinación del IDE con un entorno de prueba que agiliza la seguridad en la calidad, integración y las etapas de los procesos. Brinda todo lo que se necesita para construir, probar y entregar aplicaciones de alto rendimiento. Contiene distintas funcionalidades como generación de código, formateo de código configurable, pruebas unitarias y desarrollo remoto (STUDIO, 2014). Aunque cuenta con estas valiosas características, se requiere de un costo de adquisición de 299€ (Zend, 2015).
- **NetBeans:** es multiplataforma (Windows, Linux, Mac OS X y Solaris), gratuito, de código abierto (con licencia CDDL) y se puede utilizar para programas en otros lenguajes además de PHP. A parte de las funciones básicas con las que debería contar cualquier IDE, como resaltado de sintaxis, autocompletado, formateo de código o depurador (xDebug), también cuenta con otras funcionalidades menos comunes como la integración con PHPUnit para las pruebas unitarias y con CVS, Subversion y Mercurial para el control de versiones (MICROSYSTEMS, 2016).
- **PhpStorm:** *JetBrains PhpStorm* es un IDE multiplataforma comercial para PHP desarrollado sobre la plataforma *JetBrains IntelliJ IDEA*. Este sistema cuenta con un grupo de funcionalidades similares al NetBeans, pero la licencia personal cuesta 88€ y la comercial 176€. Proporciona un editor para PHP, HTML y *JavaScript* con el análisis de código sobre la marcha, la prevención de errores y refactorizaciones automáticas para PHP y *JavaScript*. Ofrece soporte para PHP5, incluyendo generadores, lista en *foreach*, espacios de nombres, los cierres, los rasgos y la sintaxis de matrices. Además, incluye un editor de pleno derecho de SQL con resultados de las consultas en modo editables (JetBrains, 2016).

Atendiendo al estudio anterior, el IDE para la implementación recomendado es NetBeans debido a que permite un desarrollo rápido y fácil de aplicaciones Java de escritorio y móviles, además de aplicaciones web. También facilita la construcción de aplicaciones HTML5 con HTML, JavaScript y CSS. Este IDE también proporciona un gran conjunto de herramientas para desarrolladores de PHP y C / C ++

(MICROSYSTEMS, 2016). Es gratuito y de código abierto y se encuentra respaldado por una gran comunidad de usuarios y desarrolladores de todo el mundo.

1.6. Herramientas y tecnologías para el desarrollo

A continuación, se definen los *framework* de trabajo y lenguajes de programación *web* a utilizar para el desarrollo, así como el entorno de desarrollo que facilitará el uso de estos. También se describe el Sistema Gestor de Base de Datos (SGBD) y el servidor *web* que permitirán la correcta ejecución de la aplicación. Por último, se especifica la herramienta CASE que facilitará la elaboración de los diagramas y representaciones que resulten necesarias en el proceso de desarrollo.

HTML5:

HTML son las siglas de *Hypertext Markup Language* (HTML), es un lenguaje diseñado para crear páginas *web* y otros documentos que sean posibles visualizar en el navegador *web*, es esencial para la creación de páginas y mostrar el contenido en ellas. El HTML se encarga de desarrollar una descripción sobre los contenidos que aparecen como textos y sobre su estructura, complementando dicho texto con diversos objetos (como imágenes, animaciones, videos) (Pavan, 2013). HTML5 no es una nueva versión del antiguo lenguaje de etiquetas, ni siquiera una mejora de la ya antigua tecnología, sino un nuevo concepto para la construcción de sitios web y aplicaciones en una era que combina dispositivos móviles, computación en la nube y trabajos en red.

CSS3:

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. Permite definir los colores, el diseño, y las fuentes a utilizar en la web. También posibilita adaptar la presentación a los diferentes tipos de dispositivos, como grandes pantallas, pantallas pequeñas o impresoras. CSS es independiente de HTML y se puede utilizar con cualquier lenguaje de marcado basado en XML. Resulta ser la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas (EGUILUZ, 2009).

JavaScript:

Es un lenguaje de programación interpretado que se utiliza principalmente para crear páginas web dinámicas. El lenguaje ya ha sido adoptado por la mayoría de los principales navegadores *web* gráficos. Ha hecho posible el desarrollo de aplicaciones *web* modernas con las que se puede interactuar directamente, sin hacer una recarga de la página para cada acción. Pero también se utiliza en sitios *web* más tradicionales para proporcionar diversas formas de interactividad y formas inteligentes (VALDÉS, 2007).

PHP5:

Acrónimo recursivo de *Hypertext Preprocessor* (PHP), es un lenguaje de código abierto muy popular, especialmente adecuado para el desarrollo *web* y que puede ser incrustado en HTML. Está enfocado principalmente a la programación de *scripts* del lado del servidor. PHP explota con eficiencia las principales características para seleccionar un lenguaje de *script*, ellas son: velocidad, estabilidad, seguridad y simplicidad. Es completamente expansible, está compuesto de un sistema principal, un conjunto de módulos y una variedad de extensiones de código. Además, dispone de una amplia gama de librerías que le permiten ser utilizado en muchas áreas diferentes, tales como encriptado, gráficos y XML; así como interactuar con varios motores de bases de datos, tales como MySQL, MSSQL, Oracle, Informix, PostgreSQL, entre otros *frameworks* de desarrollo (Meriño, 2008).

Bootstrap:

Bootstrap es un potente framework de CSS que fue creado por *Twitter* para simplificar el proceso de maquetación *web responsive*. Sin tener grandes conocimientos en CSS, con esta herramienta es posible comenzar a maquetar sitios *web* adaptables a todo tipo de dispositivos.

Bootstrap, es un framework originalmente creado por *Twitter*, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice a partir del diseño *responsive*. Posee numerosos componentes que facilitan el trabajo con la utilización de estándares como HTML5 y CSS3; además funciona en todos los navegadores *web*, incluido Internet Explorer (Rodríguez, 2012).

jQuery:

Es una biblioteca JavaScript rápida, pequeña y rica en funciones. Permite simplificar la manera de interactuar y manipular documentos HTML, manejar eventos, permite desarrollar animaciones y agregar

interacciones con la técnica AJAX a páginas web. jQuery una tecnología libre y de código abierto, posee un doble licenciamiento bajo la licencia MIT y la Licencia Pública General de GNU V2, por lo que es posible su uso en proyectos libres y privados. Con una combinación de versatilidad y extensibilidad, jQuery ha cambiado la forma en que millones de desarrolladores escriben código *JavaScript* (Foundation, 2016).

MySQL:

Es un SGBD para bases de datos relacionales que utiliza múltiples tablas para almacenar y organizar la información. MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos. También es muy destacable, la condición de código abierto de MySQL, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad. Esto ha favorecido de forma positiva en su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a Internet (Casillas Santillán, y otros, 2008). La API se encuentra disponible para su utilización desde los lenguajes C, C++, Eiffel, Java, Perl, TCL, Python, Ruby y PHP. También resulta destacable su velocidad de respuesta y garantiza además seguridad de acceso a datos a partir de administración basada en usuarios y privilegios.

Servidor web Apache:

Apache es un proyecto de la Fundación de Software Apache, con el objetivo de suministrar un servidor seguro, eficiente, y extensible que proporcione servicios HTTP en sincronía con los estándares HTTP actuales.

Como servidor *web* se utilizará Apache Server. Es una herramienta de código abierto y su desarrollo comenzó en febrero de 1995, por Rob McCool, en una tentativa de mejorar el servidor existente en el *National Center for Supercomputing Applications* (NCSA). La primera versión apareció en enero de 1996, el Apache 1.0. Hacia el 2000, el servidor Web Apache era el más extendido en el mundo. El nombre «Apache» es un acrónimo de «*a patchy server*» -un servidor de remiendos-, es decir un servidor construido con código preexistente y piezas y parches de código. Es la auténtica «*kill app*» del *software* libre en el ámbito de los servidores y el ejemplo de *software* libre de mayor éxito, por delante incluso del *kernel* Linux. Desde hace años, más del 60% de los servidores web de Internet emplean Apache (MASTERMAGAZINE, 2014).

Visual Paradigm v8.0:

Es una herramienta CASE que utiliza para el modelado el lenguaje Unified Model Language (UML, por sus siglas en inglés). Es libre y multiplataforma, además se caracteriza por su robustez, usabilidad y portabilidad. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Se integra fácilmente con varios entornos de desarrollo y permite la generación semiautomática de código a partir de los diagramas construidos. Esta herramienta permite realizar ingeniería tanto directa como inversa, a partir de un modelo relacional en SQL Server, MySQL, PostgreSQL, es capaz de desplegar todas las clases asociadas a las tablas (siguiendo el patrón de diseño Una Clase-Una Tabla). Soporta múltiples usuarios trabajando sobre el mismo proyecto, permite control de versiones y genera de forma automática documentación en formatos web y PDF (VisualParadigm.com, 2016).

1.7. Conclusiones

A partir del estudio de aplicaciones orientadas al análisis del rendimiento de los deportistas en las tareas que realizan como jugadores y la observación de la gestión de esta información, fue posible arribar a las siguientes conclusiones:

- Se selecciona como IDE la herramienta NetBeans en su versión 8.0.2 porque permite la construcción estructurada, escalable y extensible de aplicaciones *web*, optimizando el empleo de lenguajes como HTML5, CSS3, *JavaScript* y PHP5. Permite además minimizar el tiempo de escritura y depuración de códigos.
- Como complemento para garantizar el correcto funcionamiento de la aplicación, se selecciona el paquete de programas *AppServer*, que incluye el *Apache Server* como servidor *web*, PHP como lenguaje de programación y MySQL como SGBD.
- Para llevar a cabo una correcta ingeniería de *software* se acoge XP como metodología de desarrollo, ya que garantiza la generación de artefactos que son necesarios para una mayor comprensión de la solución a desarrollar. Complementando el proceso de ingeniería con la utilización del lenguaje de modelado UML y la herramienta CASE Visual Paradigm v8.0.

CAPÍTULO 2: SOLUCIÓN PROPUESTA

En el presente capítulo se presenta una propuesta de solución para los problemas detectados en el estudio del estado del arte realizado en el Capítulo 1. Se mencionan los roles que intervienen en la solución. También se definen las Historias de Usuario (HU), la planificación de entrega de versiones del producto y el diseño del sistema de acuerdo a las fases que propone XP.

2.1. SISFUT

SISFUT es un sistema *web* para la gestión de torneos de fútbol en la UCI. Este sistema se apoya principalmente en la información recogida antes del comienzo del torneo, dígame: datos de los jugadores, los equipos que van a participar, los árbitros, el calendario, los directores técnicos, el calendario propuesto, además de la información recogida en cada uno de los partidos. Esta herramienta cuenta además con un módulo para entrenadores, el cual le permite enfrentar dos equipos y estimar la posible alineación regular de uno de ellos atendiendo a informaciones obtenidas a partir de los enfrentamientos entre ellos. Además, ofrece información sobre la actuación de cada uno de los jugadores de cada equipo y aplicando procesamiento basado en caso, la aplicación es capaz de realizar la estimación antes mencionada, por lo que se ofrece un potencial apoyo a la toma de decisiones de los entrenadores. SISFUT cuenta con una página informativa orientada a los seguidores de cada uno de los equipos, donde se muestran datos estadísticos de todos los torneos con sus ediciones entre los que se pueden mencionar: listado de equipos que participan en un torneo y una edición específica, la plantilla oficial de ese equipo con el resumen de su participación en el torneo (partidos jugados, goles, asistencias, tarjetas amarillas, tarjetas rojas); líderes en las categorías: asistencias, goles, tarjetas amarillas, tarjetas rojas. También permite visualizar la tabla de clasificación del torneo seleccionado, calendario de la competición por jornadas con un resumen del partido en caso de que haya finalizado, galería de imágenes del torneo, así como anuncios de próximos eventos o partidos.

Roles del sistema

El sistema cuenta con autenticación basada en roles, la cual brindará servicios en dependencia del tipo de usuario autenticado, esto permite mantener la seguridad de los datos que se manejan. El administrador es

el que tiene todos los privilegios sobre las acciones que ofrece la aplicación. A continuación, se muestra un listado con los roles y permisos que tendrá cada usuario.

Tabla 1: Roles del sistema

Roles	Descripción
Administrador	Responsable de mantener el funcionamiento correcto de la herramienta. Posee permisos para ejecutar todas las funciones administrativas de la plataforma y es el encargado de gestionar la información. Establece los permisos pertinentes para los diferentes roles de usuarios.
Estadísticos	Posee privilegios para administrar las noticias relacionadas con los eventos que se realizan en la Universidad. Además es el encargado de introducir las acciones que realiza cada jugador en el partido.
Entrenador	Consultará las estadísticas que le proporcione el sistema.

Para que el sistema pueda realizar la estimación del posible equipo regular que alineará el próximo partido, el sistema define el siguiente grupo de variables correspondientes a las acciones de los jugadores, así como la dependencia funcional que existe entre estas.

Variables analizadas para la toma de decisiones

Durante un partido el entrenador y sus asistentes tienen en cuenta una serie de acciones que son registradas para un análisis posterior, en búsqueda de mejorar el entrenamiento, las tácticas, técnicas y nivel de juego del equipo. Las principales acciones que a nivel de jugador individual y de colectivo son tabuladas para su análisis y toma de decisiones son las siguientes:

- (Pe) Penalti.
- (Dp) Doble penalti.
- (A) Asistencia.
- (TA) Tarjeta amarilla.
- (TR) Tarjeta roja.

- (/R) Interceptado por un regate.
- (TP) Tiro a portería.
 - (Tg) gol.
 - (T+) tiro positivo.
 - (T-) tiro negativo.
- (TLD) Tiro libre directo.
- (TLI) Tiro libre indirecto.
- (Rb) Recuperación del balón.

Datos individuales ofensivos:

- (PPO+) Pases positivos ofensivos: pases que se realizan a la zona ofensiva de ataque y son correctamente ejecutados.
- (PNO-) Pases negativos ofensivos: pases que se realizan a la zona ofensiva de ataque y son interceptados, anticipados o mal ejecutados.
- (PAG+) Pases asistencia a gol: pases que se realizan a la zona ofensiva de ataque y el jugador define gol.
- (PAG-) Pases asistencia a gol: pases que se realizan a la zona ofensiva de ataque y el jugador no define gol.

Datos individuales defensivos:

- (IP+) Interceptación positiva: es la que se realiza cuando el jugador se queda con el control del balón.
- (IN-) Interceptación negativa: es la que se realiza mediante un despeje y el jugador no se queda con ella o es bote o interceptada o anticipada, por el contrario.
- (AP+) Anticipación positiva: es la que se realiza cuando el jugador se anticipa al control del balón que va a poseer el contrario por medio de un pase o despeje, manteniendo el control del mismo.
- (AN-) Anticipación negativa: es la que se realiza cuando el jugador se anticipa al control del balón que va a poseer el contrario por medio de un pase o despeje y al anticipar no mantiene el control del balón.

La mayoría de estas variables se definen para evaluar el desempeño de cada jugador en los partidos de futsal, luego se consulta toda la información para poder ofrecer la funcionalidad de estimar los 5 mejores jugadores de un equipo. Con esta predicción se apoya a la toma de decisiones para celebrar el próximo partido de una jornada deportiva. Para realizar una estimación lo más acertada posible se implementan procedimientos que se corresponden con el razonamiento basado en casos, en específico se utiliza la técnica de "sistema experto", esta técnica trata de imitar funciones de un experto en algún dominio de conocimiento, es decir, manipulan conocimientos de expertos para resolver eficiente o efectivamente, problemas en un área específica, tal como hacen los expertos humanos (RUSSELL, S. J.: PÍORVIG, P., 2004). Una vez que el entrenador selecciona su equipo, así como el equipo que va a enfrentar en la próxima jornada, solicita a la aplicación predecir los 5 jugadores regulares para llevar a cabo este partido. Luego el sistema analiza la información referente a cada jugador valorando las acciones registradas en el evento que se está desarrollando a la hora del análisis. También se valora de forma adicional, las acciones de cada jugador en enfrentamientos pasados con el equipo rival, sin restringir la recuperación de datos al presente evento. A continuación, se muestra algunos de estos pesos asignados a las variables:

- Tg -> 1.0.
- Asistencia -> 0.5.
- Pe -> 0.7.
- Dp -> 0.8.
- T+ -> 0.3.
- T- -> 0.1.
- Rb -> 0.3.

2.2. Fase I: Exploración

La metodología de desarrollo XP comienza con la fase de exploración. Durante esta etapa se realiza el proceso de identificación de las HU, estas constituyen uno de los artefactos más importante que se generan en la metodología, pues son la forma en que se especifican los requisitos del sistema y además se define el alcance general del proyecto (Joskowicz, 2008).

2.2.1. Historias de Usuario

Las HU son la técnica utilizada para representar y especificar los requisitos del *software*. Se trata de un conjunto de tablas en las cuales el cliente describe de forma breve las características que el sistema debe poseer. El tratamiento de las HU es muy dinámico y flexible. Tienen el mismo propósito que los casos de uso de las metodologías tradicionales, las escriben los propios clientes tal y como ven ellos las necesidades del sistema. Las HU son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante las iteraciones que se definan (Letelier, y otros, 2013).

A continuación, se describen un grupo de HU definidas para llevar a cabo el desarrollo de SISFUT, el resto pueden ser consultadas en los anexos del trabajo.

Tabla 2: HU1 Adicionar torneo

Historia de Usuario	
Número: 1	Nombre de Historia de Usuario: Adicionar torneo
Usuario: Administrador	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 0.5
Riesgo en Desarrollo: Media	Programador responsable: Yalbert González Antúnez
<p>Descripción:</p> <p>Permite al administrador adicionar un torneo llenando los siguientes campos:</p> <ul style="list-style-type: none"> • Nombre del torneo • Ente organizador • Descripción • Fecha de Fundación • Número de la Edición • Temporada • Sistema de competición • Fecha de la edición 	
<p>Observaciones: Para poder adicionar un torneo el usuario que se registre debe contar con el rol Administrador.</p>	

Prototipo de Interfaz:

Adicionar nuevo torneo

Torneos > Adicionar torneo

Nombre del torneo

Ente organizador

Descripción

Fecha de fundación

Tabla 3: HU11 Generar principales líderes

Historia de Usuario	
Número: 11	Nombre de Historia de Usuario: Generar principales líderes
Usuario: Usuario	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1.5
Riesgo en Desarrollo: Alta	Programador responsable: Yalbert González Antúnez
Descripción: Permite observar una tabla con los resultados de los líderes en cada departamento. Estos departamentos son: goleadores, asistencias, tarjetas amarillas y tarjetas rojas.	
Observaciones: Para poder generar los líderes deben existir jugadores registrados en el sistema. No es necesario que se registre un usuario para poder acceder a esta funcionalidad.	

Prototipo de Interfaz:



Tabla 4: HU12 Generar las estadísticas finales de cada partido

Historia de Usuario	
Número: 12	Nombre de Historia de Usuario: Generar las estadísticas finales de cada partido.
Usuario: Entrenador	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1.5
Riesgo en Desarrollo: Alta	Programador responsable: Yalbert González Antúnez
Descripción:	
Permite a los usuarios con rol Entrenador acceder a una vista que, en correspondencia con un evento competitivo, una edición y dos equipos seleccionados, muestra una tabla con los diferentes resultados como por ejemplo goles, asistencias, cantidad de faltas cometidas por cada jugador; así como los jugadores titulares de ese partido y las tarjetas amarillas y rojas que estos pudieron haber recibido.	

Observaciones: Para que se muestre el resumen general de cualquier partido al entrenador, debe de haber concluido el partido en cuestión.

Prototipo de Interfaz:

Goles	<input type="text" value="4"/>		<input type="text" value="0"/>	Goles
Disparos a Puerta	<input type="text" value="4"/>		<input type="text" value="0"/>	Disparos a Puerta
Disparos in 3 Palos	<input type="text" value="4"/>		<input type="text" value="0"/>	Disparos in 3 Palos
Comers	<input type="text" value="0"/>		<input type="text" value="0"/>	Comers
Faltas 1er tiempo	<input type="text" value="0"/>		<input type="text" value="0"/>	Faltas 1er tiempo
Faltas 2do tiempo	<input type="text" value="0"/>		<input type="text" value="0"/>	Faltas 2do tiempo
Tarjetas amarillas	<input type="text" value="0"/>		<input type="text" value="0"/>	Tarjetas amarillas
Tarjetas rojas	<input type="text" value="0"/>		<input type="text" value="0"/>	Tarjetas rojas
Minuto 1er tiempo	<input type="text" value="0"/>		<input type="text" value="0"/>	Minuto 1er tiempo
Minuto 2do tiempo	<input type="text" value="0"/>		<input type="text" value="0"/>	Minuto 2do tiempo

Tabla 5: HU13 Generar los 5 mejores jugadores de un equipo

Historia de Usuario	
Número: 13	Nombre de Historia de Usuario: Generar los 5 mejores jugadores de un equipo.
Usuario: Entrenador	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1.5
Riesgo en Desarrollo: Alta	Programador responsable: Yalbert González Antúnez
Descripción:	

Permite que el entrenador pueda consultar, dado un evento competitivo, una edición y dos equipos; los 5 mejores jugadores del primer equipo para jugar en la próxima fecha del evento seleccionado. Para ello se valoran las acciones de cada jugador del primer equipo de acuerdo a sus actuaciones en el evento; además se tienen en cuenta los enfrentamientos en otros eventos con el segundo equipo analizado.

Observaciones: Para obtener una propuesta eficiente debe existir el evento, la edición y el equipo a evaluar para determinar los mejores 5 jugadores.

Prototipo de Interfaz:

Tabla 6: HU14 Generar evento competitivo

Historia de Usuario	
Número: 14	Nombre de Historia de Usuario: Generar evento competitivo.
Usuario: Usuario	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1.5
Riesgo en Desarrollo: Alta	Programador responsable: Yalbert González Antúnez
Descripción: Permite acceder a una vista que muestra una tabla con los resultados de todos los partidos realizados en el evento competitivo seleccionado.	
Observaciones: Para que el usuario pueda observar los resultados del evento, este debe existir y además, se deben tener registros de los partidos celebrados.	

Prototipo de Interfaz:

2.3. Fase II: Planificación

En esta fase el cliente establece la prioridad de cada HU y en correspondencia, los programadores realizan una estimación del esfuerzo necesario para realizar cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente (Letelier, y otros, 2013).

2.3.1. Lista de reservas del producto

La lista de reservas del producto es una tabla que recoge los requisitos funcionales y no funcionales de una aplicación, el tiempo de duración estimado en semanas para el desarrollo de cada funcionalidad y el rol que realiza la estimación. A continuación, se muestra:

Tabla 7: *Lista de reservas del producto*

Ítem*	Descripción	Estimación	Estimado por:
Prioridad: Alta			
1	Adicionar torneo.	0.5	Analista.
2	Adicionar edición.	0.5	Analista.
3	Adicionar equipo.	0.5	Analista.
4	Adicionar jornada.	0.5	Analista.
5	Adicionar calendario.	0.5	Analista.

6	Adicionar jugadores.	0.5	Analista.
7	Adicionar contrato.	0.5	Analista.
8	Adicionar árbitro.	0.5	Analista.
9	Adicionar director técnico.	0.5	Analista.
10	Adicionar noticia.	0.5	Analista.
11	Generar principales líderes.	1.5	Analista.
12	Generar las estadísticas finales de cada partido.	1.5	Analista.
13	Generar los 5 mejores jugadores de un equipo.	1.5	Analista.
14	Generar evento competitivo.	1.5	Analista.
Prioridad: Media			
15	Modificar torneo.	0.2	Analista.
16	Modificar edición.	0.2	Analista.
17	Modificar equipo.	0.2	Analista.
18	Modificar jornada.	0.2	Analista.
19	Modificar calendario.	0.2	Analista.
20	Modificar jugador.	0.2	Analista.
21	Modificar contrato.	0.2	Analista.
22	Modificar árbitro.	0.2	Analista.
23	Modificar directores técnicos.	0.2	Analista.
24	Modificar noticia.	0.2	Analista.
25	Listar calendario.	0.1	Analista.

26	Eliminar jugador.	0.1	Analista.
Prioridad: Baja			
27	Eliminar torneo.	0.1	Analista.
28	Eliminar edición.	0.1	Analista.
29	Eliminar equipo.	0.1	Analista.
30	Eliminar jornada.	0.1	Analista.
31	Eliminar calendario.	0.1	Analista.
32	Eliminar contrato.	0.1	Analista.
33	Eliminar árbitro.	0.1	Analista.
34	Eliminar directores técnicos.	0.1	Analista.
35	Eliminar noticia.	0.1	Analista.
36	Listar torneos.	0.1	Analista.
37	Listar ediciones.	0.1	Analista.
38	Listar equipos.	0.1	Analista.
39	Listar jornadas.	0.1	Analista.
40	Listar jugadores.	0.1	Analista.
41	Listar contratos.	0.1	Analista.
42	Listar árbitros.	0.1	Analista.
43	Listar directores técnicos.	0.1	Analista.
44	Listar noticias.	0.1	Analista.

Tabla 8: *Requisitos no funcionales.*

Requisitos No Funcionales:		
1	<p>Usabilidad: permitir una navegación sencilla, esto se logrará a partir de una estructura de la información correcta, en todo momento los usuarios tendrán conocimiento del lugar donde se encuentra en el sistema a través del uso de títulos, subtítulos. Los enlaces y botones serán fáciles de asociar con las operaciones que realizan.</p>	Analista.
2	<p>Apariencia o interfaz externa: la interfaz gráfica debe ser uniforme, incluyendo pantallas, menús y opciones. La consistencia de la interacción entre usuario y componente estará determinada por el diseño de la interfaz de usuario que mantendrán los elementos como menús y zonas de trabajo, en posiciones fijas, además de la mayor uniformidad posible entre cuadros de texto y botones. Tanto los títulos de los componentes de la interfaz, como los mensajes para interactuar con los usuarios, así como los mensajes de error, deberán ser en idioma español.</p>	Analista.
3	<p>Confiabilidad: ante cualquier evento, ya sea correcto o incorrecto, se mostrará un mensaje informativo. Frente a la falta de fluido eléctrico el sistema reinicia el proceso que desarrollaba en momentos del incidente.</p>	Analista.
4	<p>Portabilidad: Permitir que la aplicación se ejecute en sistemas operativos como GNU/Linux y Windows, garantizando un producto multiplataforma.</p>	Analista.
5	<p>Seguridad: Permitir el acceso a la información de acuerdo a los permisos y rol del usuario, teniendo en cuenta en todo momento los objetivos y principios de la seguridad informática. Los usuarios deben autenticarse antes de interactuar con él, para garantizar un acceso a la información controlado.</p> <p>Este deberá garantizar el acceso controlado a la información. Este debe influir sobre cómo se presentan las interfaces para cada usuario dependiendo del nivel de acceso a la información con que cuente.</p>	Analista.

6	Eficiencia: Los tiempos de respuesta y velocidad de procesamiento de la información no deben ser mayores de 3 segundos para las actualizaciones y 10 para las recuperaciones.	Analista.
---	--	-----------

2.3.2. Desarrollo del Plan de duración de las Iteraciones

Luego de identificar las HU y haber realizado la estimación del esfuerzo por cada una de ellas, se procede a realizar el plan de iteraciones, en el cual estarán contenidas las HU en el orden a realizar por cada iteración según su nivel de prioridad, así como la descripción y el total de semanas que durarán cada una de estas. A continuación, se muestra el plan de iteraciones:

Tabla 9: Plan de duración de las Iteraciones

Iteración	Descripción	Orden de la HU a implementar	Duración total
1	En esta primera iteración se tiene como principal objetivo implementar las HU que poseen prioridad “Alta”, las cuales representan funcionalidades sensibles del sistema que inciden críticamente en el funcionamiento del mismo.	1 – 14.	11.0
2	En esta iteración se implementaran las HU que cuentan con una prioridad “Media”, al igual que las anteriores son de gran importancia en cuanto a su incidencia en el funcionamiento del producto final pero muchas dependen de la implementación de las HU anteriores.	15 – 26.	2.2
3	En esta última iteración se implementarán las HU que cuentan con una prioridad “Baja”, que, aunque cuenten	27 – 44	1.8

	con esta prioridad, son importantes para garantizar un funcionamiento completo de la aplicación.		
--	--	--	--

2.3.3. Plan de Entregas

A partir del plan de iteraciones descritas anteriormente se procede a realizar el Plan de Entregas, el cual tiene como objetivo dar a conocer las fechas de culminación de las iteraciones y sus correspondientes HU.

Tabla 10: *Plan de Entregas*

Iteración	1	2	3
Cantidad de HU	14	12	18
Fecha de Entrega	18-01-2016	05-02-2016	15-04-2016

2.4. Fase III: Iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de cuatro semanas. Al final de la última iteración el sistema estará listo para entrar en producción (JOSÉ H. CANÓS, P. L., M^a CARMEN PENADÉS., 2002).

Diseño del sistema:

La metodología XP no requiere de la representación del sistema mediante diagramas de clases utilizando UML para diseñar las aplicaciones, en este caso utiliza otras técnicas como las tarjetas CRC (Cargo o clase, Responsabilidad, Colaboración).

2.4.1. Tarjetas CRC

Las tarjetas CRC permiten que el equipo completo contribuya en la tarea del diseño. Como su nombre lo indica se dividen en tres secciones: el nombre de la clase, sus responsabilidades y sus colaboradores. El nombre de la clase se coloca a modo de título en la tarjeta, las responsabilidades se colocan a la izquierda y las clases que se implican en cada responsabilidad a la derecha. Una clase describe cualquier objeto o

evento, mediante los atributos y los métodos, las responsabilidades son las tareas que realizan o los métodos correspondientes a la clase y los colaboradores son las demás clases con las que trabaja conjuntamente para cumplir con sus responsabilidades (Escribano, 2002). Se describen las principales tarjetas CRC, las demás se pueden consultar en los anexos del trabajo.

Tabla 11: Tarjeta CRC insertarTorneos

Tarjeta CRC	
Clase: insertarTorneos	
Responsabilidades:	Colaboradores:
1-Adiciona un nuevo torneo. 2-Busca por el identificador cualquier torneo creado. 3-Lista todos los torneos creados. 4-Modifica un torneo creado. 5-Elimina un torneo creado.	- insertarEdiciones

Tabla 12: Tarjeta CRC insertarPartido

Tarjeta CRC	
Clase: insertarPartido	
Responsabilidades:	Colaboradores:

<p>1-Adiciona un nuevo partido. 2-Busca por el identificador cualquier partido creado. 3-Lista todos los partidos creados. 4-Modifica un partido creado. 5-Elimina un partido creado.</p>	<p>- insertarJornadas</p>
---	---------------------------

Tabla 13: Tarjeta CRC insertarArbitros

Tarjeta CRC	
Clase: insertarArbitros	
Responsabilidades:	Colaboradores:
<p>1-Adiciona un nuevo árbitro. 2-Busca por el identificador cualquier árbitro creado. 3-Lista todos los árbitros creados. 4-Modifica un árbitro creado. 5-Elimina un árbitro creado.</p>	<p>- insertarEquipos</p>

2.5. Arquitectura de software

Según el estándar IEEE (*Institute of Electrical and Electronics Engineers*): “La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución”.

2.5.1. Patrón Modelo Vista Controlador (MVC)

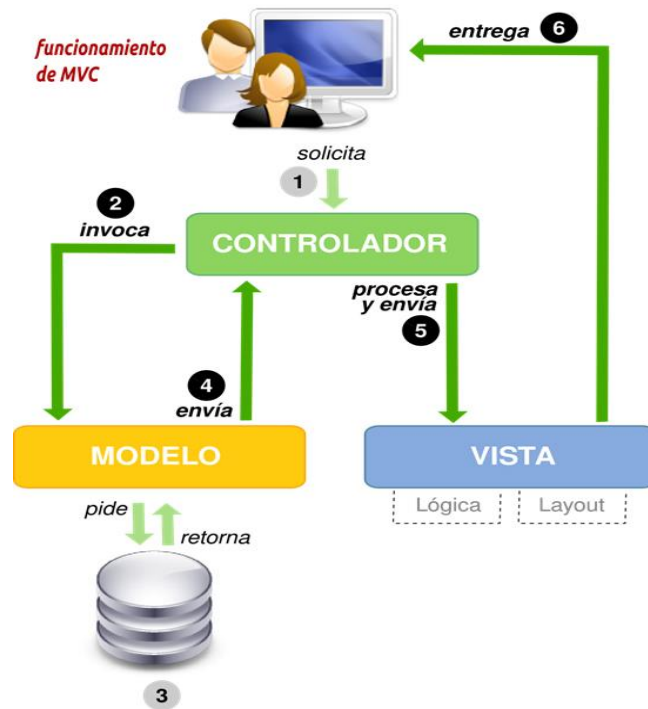


Figura 2: Representación del funcionamiento del patrón MVC

El Modelo es la representación de los datos que maneja la aplicación. Este no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el responsable de acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento. También se encarga de manejar las reglas del negocio.

La Vista es el elemento que maneja la presentación visual de los datos representados por el Modelo. Es el medio de comunicación entre los usuarios y los controladores, por esta vía los usuarios realizan las peticiones e ingresan los datos necesarios para obtener los resultados que arrojan los procesos administrativos.

El Controlador es el elemento que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo. Maneja y entrega las solicitudes del usuario, procesando la información necesaria y modificando el Modelo en caso de ser necesario. Es responsable de recibir los eventos y datos de entrada, que serán utilizados para aplicar los mecanismos que atienden el negocio y para ello utilizan las funcionalidades ofrecidas por los modelos.

Para obtener una correcta implementación se hace necesario tener en cuenta las mejores soluciones conocidas, con el objetivo de diseñar las clases y las funcionalidades a codificar. Para ello se analizan y proponen algunos patrones de diseño que responden ante las necesidades de desarrollo.

2.6. Patrones de diseño

Un patrón de diseño es una solución repetible a un problema recurrente en el diseño de *software*. Esta solución no es un diseño terminado que puede traducirse directamente a código, sino más bien una descripción sobre cómo resolver el problema, la cual puede ser utilizada en diversas situaciones. Los patrones de diseño reflejan todo el rediseño y recodificación que los desarrolladores han ido haciendo a medida que intentaban conseguir mayor reutilización y flexibilidad en su *software*. Los patrones documentan y explican problemas de diseño, para luego discutir una buena solución a dicho problema. Con el tiempo, los patrones comienzan a incorporarse al conocimiento y experiencia colectiva de la industria del *software*, lo que demuestra que el origen de los mismos radica en la práctica misma más que en la teoría (Letelier, y otros, 2013) .

2.6.1. Patrones GRASP

Los Patrones de Software para la Asignación General de Responsabilidad (GRASP, por sus siglas en inglés) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. En la implementación de la aplicación se utilizaron los siguientes:

Creador: Ayuda a identificar quién debe ser el responsable de la creación o instanciación de nuevos objetos o clases. Tiene la información necesaria para realizar la creación del objeto que es una de las actividades más comunes en un sistema orientado a objetos. También se encarga de almacenar o manejar varias instancias de la clase. En la Figura 3 se muestra el uso del patrón creador, donde la **clase_insertarTorneos** es la responsable de tener el control acerca de los torneos.

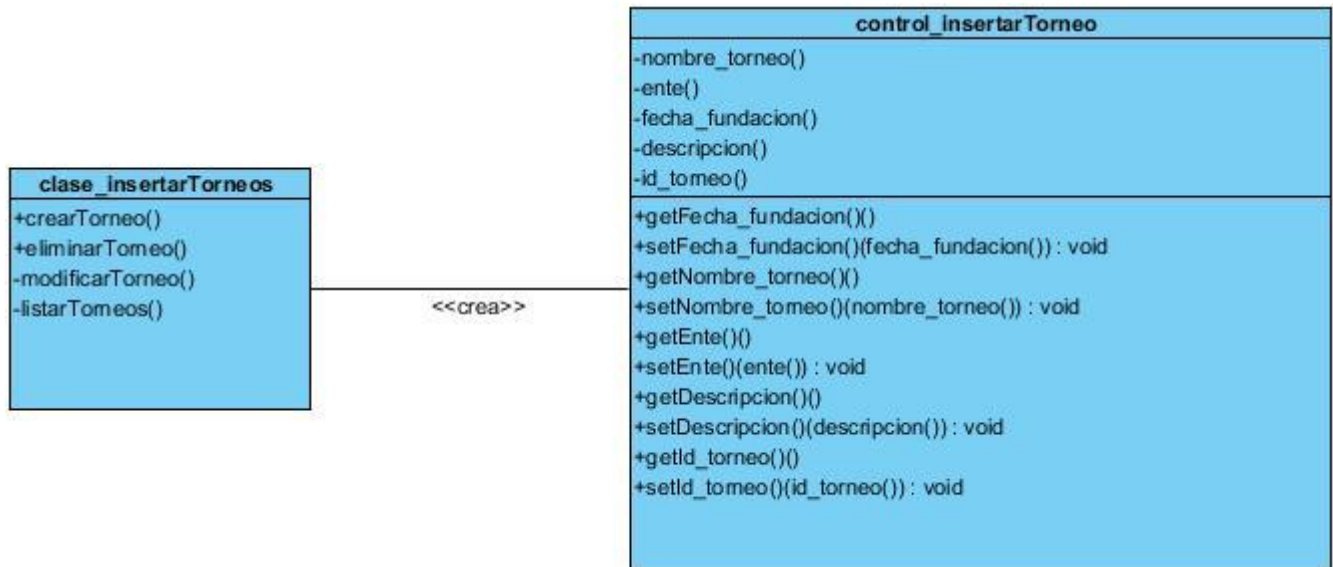


Figura 3: Diagrama de clases que representa el patrón Creador

Controlador: Es el encargado de asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. La mayor parte de los sistemas reciben eventos de entrada externa, en cualquiera de los casos que puedan existir, si se recurre a un diseño orientado a objetos, hay que elegir los controladores que manejen esos eventos de entrada.

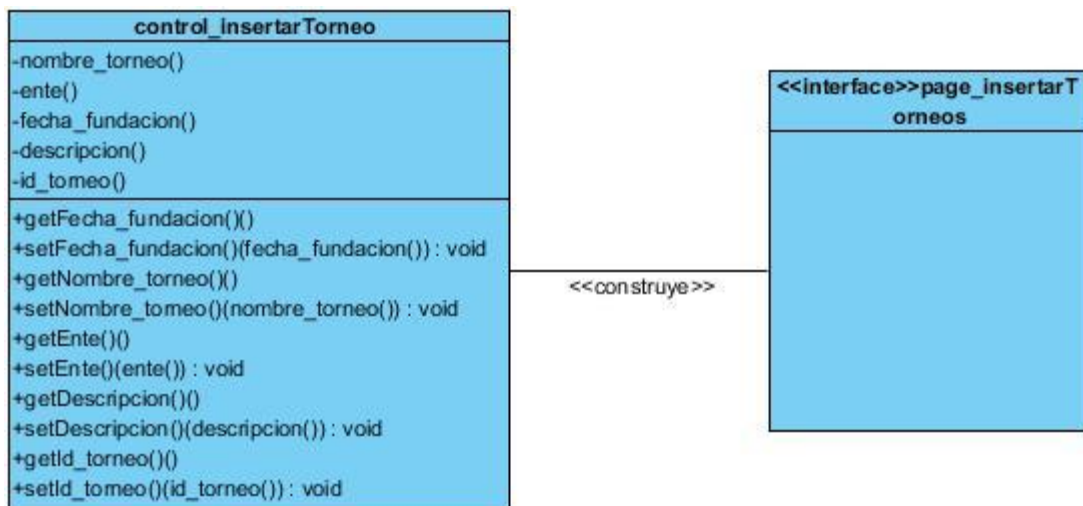


Figura 4: Diagrama de clases que representa el patrón Controlador

Experto: Es un principio básico que suele utilizarse en el diseño orientado a objetos. Expresa que siempre se debe asignar una responsabilidad al experto en información, o sea, a la clase que cuenta con la información necesaria para llevar a cabo la funcionalidad (LARMAN, 2003).

La evidencia del uso de este patrón se puede observar en la mayoría de las clases controladoras del sistema. Un ejemplo de ella podría ser **control_insertarJugadores**, que es el encargado de controlar la información respecto a los jugadores y a todos los datos relacionados con los mismos, y de esta forma, optimiza y organiza las responsabilidades sobre la información sensible de los jugadores registrados en el sistema.

2.7. Modelo de datos

El Modelo de Datos (MD) es una definición lógica, independiente y abstracta de objetos, operadores que en conjunto ayudan a diseñar aplicaciones cuyo objetivo es permitir describir y manipular información almacenada en la base de datos. El MD estudia los datos independientemente del procesamiento que los transforma. Se compone de tres piezas de información interrelacionadas: el objeto de datos, los atributos que describen el objeto de datos y la relación que conecta objetos de datos entre sí (DATE, 2003). En el Anexo IV, se visualiza el MD perteneciente a la solución propuesta.

2.8. Conclusiones del capítulo

En este capítulo se han abordado los aspectos referentes a la concepción del producto a desarrollar y sus características funcionales. Esto permitió arribar a las siguientes conclusiones:

- A partir de la definición de las HU se pudieron identificar las principales funcionalidades a desarrollar para el correcto funcionamiento de SISFUT.
- La estimación del tiempo para la implementación de las HU definidas, permitió calcular una entrega final del producto en un tiempo estimado de tres meses y medio aproximadamente.
- Fue posible identificar los patrones GRASP para el desarrollo de la solución.
- Se exponen los artefactos generados en las fases de Exploración y Planificación que establece la metodología XP.

Con los aspectos arquitectónicos y de diseño a utilizar definidos, queda establecida la vía para llevar a cabo la implementación de la aplicación web que dará solución al problema de la presente investigación.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN

En el presente capítulo se describirán las tareas de ingeniería definidas para cada historia de usuario. Se realizará la presentación de los casos de prueba creados para validar las funcionalidades que responden a los requisitos identificados en la fase de exploración y planificación.

3.1. Fase IV: Implementación

En esta fase, XP plantea que las HU seleccionadas para ser implementadas se realizan durante el transcurso de la iteración a la que pertenecen. Por estas razones, se lleva a cabo una revisión del plan de iteraciones y se modifican en caso de ser necesario. Como parte de este plan se descomponen las HU en tareas de ingeniería (Joskowicz, 2008).

3.1.1. Tareas de Ingeniería

Las tareas de ingeniería pueden estar descritas por un lenguaje técnico y no ser necesariamente entendibles por el cliente. Tienen como objetivo definir cada una de las actividades que dan cumplimiento a las HU, de forma tal que se entienda lo que el sistema tiene que hacer y facilite su construcción. Se describen algunas de las tareas de ingeniería correspondientes a las HU del sistema, el resto pueden ser consultadas en los anexos del documento.

Tabla 14: Tarea de ingeniería #1.

Tarea	
Número de tarea: 1	Número de HU: 1,15,27,36
Nombre de la tarea: Gestionar torneos	
Programador responsable: Yalbert González Antúnez	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 2 de diciembre del 2015	Fecha de fin: 9 de diciembre del 2015

Descripción: Cuando se termina la tarea le permite al administrador poder insertar un torneo, eliminar un torneo creado anteriormente, modificar cualquier torneo creado o listar todos los torneos que hayan sido creados.

Tabla 15: Tarea de ingeniería #2

Tarea	
Número de tarea: 2	Número de HU: 2,16,28,37
Nombre de la tarea: Gestionar ediciones	
Programador responsable: Yalbert González Antúnez	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 10 de diciembre del 2015	Fecha de fin: 16 de diciembre del 2015
Descripción: Cuando se termina la tarea le permite al administrador poder insertar una edición, eliminar una edición creada anteriormente, modificar cualquier edición creada o listar todas las ediciones que hayan sido creadas.	

Tabla 16: Tarea de ingeniería #3

Tarea	
Número de tarea: 3	Número de HU: 3,17,29,38
Nombre de la tarea: Gestionar equipos	
Programador responsable: Yalbert González Antúnez	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 17 de diciembre del 2015	Fecha de fin: 24 de diciembre del 2015
Descripción: Cuando se termina la tarea le permite al administrador poder insertar un equipo, eliminar un equipo creado anteriormente, modificar cualquier equipo creado o listar todos los equipos que hayan sido creados.	

Tabla 17: Tarea de ingeniería #4

Tarea	
Número de tarea: 4	Número de HU: 4,18,30,39
Nombre de la tarea: Gestionar jornadas	
Programador responsable: Yalbert González Antúnez	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 8 de enero del 2016	Fecha de fin: 15 de enero del 2016
Descripción: Cuando se termina la tarea le permite al administrador poder insertar una jornada, eliminar una jornada creada anteriormente, modificar cualquier jornada creada o listar todas las jornadas que hayan sido creadas.	

Tabla 18: Tarea de ingeniería #5

Tarea	
Número de tarea: 5	Número de HU: 5,19,31,25
Nombre de la tarea: Gestionar calendarios	
Programador responsable: Yalbert González Antúnez	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 22 de enero del 2016	Fecha de fin: 29 de enero del 2016
Descripción: Cuando se termina la tarea le permite al administrador poder insertar un calendario, eliminar un calendario creado anteriormente, modificar cualquier calendario creado o listar todos los calendarios que hayan sido creados.	

Tabla 19: Tarea de ingeniería #6

Tarea	
Número de tarea: 6	Número de HU: 6,20,26,40

Nombre de la tarea: Gestionar jugadores	
Programador responsable: Yalbert González Antúnez	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 30 de enero del 2016	Fecha de fin: 6 de febrero del 2016
Descripción: Cuando se termina la tarea le permite al administrador poder insertar un jugador, eliminar un jugador creado anteriormente, modificar cualquier jugador creado o listar todos los jugadores que hayan sido creados.	

3.2. Estilos de programación

Los estilos de programación definen la estructura y apariencia física del código, lo que facilita su comprensión, mantenimiento y lectura. En la implementación de la aplicación se utilizaron diferentes estilos que se describen de la manera siguiente:

3.2.1. Definiciones de clases

Las declaraciones de clases tienen su llave de apertura una línea más abajo de la declaración y el nombre de la clase comienza siempre con dos guiones bajos y le sigue el respectivo nombre con letra minúscula como se muestra en el siguiente ejemplo:

```
class __alineacion {
    private $conexion;

    public function __construct() {
        $this->conexion = new Conexion();
    }
}
```

Figura 5: Definición de las clases

3.2.2. Definición de métodos

Los métodos de las clases están precedidos por la palabra reservada que define el encapsulamiento del método (*public*, *protected* o *private*), seguido de la palabra *function* y después el nombre del método. El nombre comienza con una letra minúscula, en caso de ser un nombre compuesto por dos o más palabras, estos comenzarán con letra mayúscula como se muestra en la siguiente figura:

```
public function listarUsuariosDelSistema() {
    $result = mysql_query("SELECT * FROM tb_user where eliminado <> 1");
    while ($fila = mysql_fetch_array($result)) {
        $datos[] = $fila;
    }
    return $datos;
}
```

Figura 6: Definición de métodos

3.2.3. Asignaciones a variables

La asignación a variables se realiza siempre dejando un espacio antes y después del signo de igualdad (=) como se puede observar en la Figura 7 mostrada a continuación.

```
$idCuerpo = $_GET['idCuerpo'];
$id_equipos = $_POST['id_equipos'];
$nombre_ct = $_POST['nombre_ct'];
$apellidos_ct = $_POST['apellidos_ct'];
$rol_cargo = $_POST['rol_cargo'];
$ci_ct = $_POST['ci_ct'];
$segundo_apellido_ct = $_POST['segundo_apellido_ct'];
$lugar_nacimiento = $_POST['lugar_nacimiento'];
$fecha_nacimiento = $_POST['fecha_nacimiento'];
$colegio = $_POST['colegio'];
```

Figura 7: Asignación a variables

3.2.4. Estructuras de control

Las estructuras de control tales como *if()*, *for()*, *foreach()*, entre otras se utilizarán colocando la llave de apertura en la misma línea de código y la de cierre en la última línea del módulo, como se puede observar en la siguiente figura:

```

foreach ($acciones as $a) {
    $cantpart = $a['cant_partidos'];
    valor de la accion
    $coef = $coeficiente[$a['accion']];

    reiteratividad de la accion
    $reit = $coef * $a['cant_acciones'];
    if (in_array($a['accion'], array(1, 2, 7, 8))) {
        $goles ++;
    }
    si jugaste contra ese equipo
    if (in_array($equipos_2, array($a['local'], $a['visit']))) {
        $coef = $coef * 0.5;
    }
}

```

Figura 8: Estructuras de control

3.2.5. Llamada a funciones

La llamada a funciones se efectuará sin dejar espacios entre el nombre de la función, los paréntesis y los parámetros. Estos últimos separados por coma en el caso de existir más de uno y dejando un espacio después de la misma, como se puede apreciar en la siguiente figura:

```

$obj->listarJugadoresTrayectoriaTorneo($id_torneo, $id_edicion, $equipos_1);
$jugador_acciones = array();

```

Figura 9: Llamada a funciones

Con las tareas de ingeniería definidas y habiendo establecido los estándares de codificación, se hace necesario establecer un conjunto de pruebas para comprobar la calidad de la solución implementada. Luego se analizan estos casos de prueba y se ejecutan, lo que permite medir el nivel de cumplimiento con los objetivos de implementación trazados y el nivel de satisfacción del cliente.

3.3. Fase V: Pruebas

Las pruebas son un conjunto de actividades que se pueden planificar por adelantado y llevar a cabo sistemáticamente. Por esta razón, se deben definir en el proceso de la ingeniería del *software*. Todo esto, contribuye a elevar la calidad de los productos desarrollados y a la seguridad de los programadores a la hora de introducir cambios o modificaciones (Joskowicz, 2008).

La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente (Escribano, 2002).

Se decide para una primera versión del producto hacer pruebas de aceptación pues así lo solicito el cliente que forma parte del equipo de desarrollo. En próximas iteraciones del producto se pretende aplicar las demás pruebas.

3.3.1. Pruebas de aceptación

Las pruebas de aceptación son especificadas por el cliente, se centran en las características y funcionalidades generales del sistema que son visibles. Estas pruebas derivan de las HU que se han implementado como parte de la liberación del software. Una prueba de aceptación es como una caja negra. Cada una de ellas representa una salida esperada del sistema. Es responsabilidad del cliente verificar la corrección y toma de decisiones acerca de estas pruebas. A continuación, se muestra una representación de las pruebas de aceptación a realizarse en las iteraciones de desarrollo, el resto pueden ser consultadas en los anexos del documento.

Tabla 20: Caso de prueba para Adicionar un torneo

Caso de Prueba de Aceptación	
Número : 1	Historia de Usuario: 1
Nombre: Adicionar torneo	
Condiciones de Ejecución: El administrador debe estar previamente autenticado.	

<p>Entradas/Pasos de ejecución:</p> <p>- El administrador llena todos los campos necesarios para poder insertar el torneo.</p>
<p>Resultados Esperados:</p> <p>Se visualiza en una vista con los datos del nuevo torneo creado.</p>
<p>Evaluación de la prueba: Satisfactoria.</p>

Tabla 21: Caso de prueba para Adicionar una edición

Caso de Prueba de Aceptación	
Número : 2	Historia de Usuario: 2
Nombre: Adicionar edición	
<p>Condiciones de Ejecución:</p> <p>a)-El administrador debe estar previamente autenticado.</p> <p>b)-Debe existir un torneo en el sistema.</p>	
<p>Entradas/Pasos de ejecución:</p> <p>- El administrador llena todos los campos necesarios para poder insertar una edición en cualquier torneo creado.</p>	
<p>Resultados Esperados:</p> <p>Se visualiza en una vista los datos de la nueva edición creada.</p>	
<p>Evaluación de la prueba: Satisfactoria.</p>	

Tabla 22: Caso de prueba para Adicionar un equipo

Caso de Prueba de Aceptación	
Número : 3	Historia de Usuario: 3

Nombre: Adicionar equipo
Condiciones de Ejecución: a)-El administrador debe estar previamente autenticado. b)- Debe existir un torneo en el sistema y una edición.
Entradas/Pasos de ejecución: - El administrador llena todos los campos necesarios para poder insertar un equipo en cualquier edición de cualquier evento creado.
Resultados Esperados: Se visualiza en una vista con los datos del nuevo equipo creado.
Evaluación de la prueba: Satisfactoria.

Tabla 23: Caso de prueba para Generar evento competitivo

Caso de Prueba de Aceptación	
Número : 11	Historia de Usuario: 11
Nombre: Generar evento competitivo	
Condiciones de Ejecución: a)-Debe existir al menos un evento creado en el sistema.	
Entradas/Pasos de ejecución: - El usuario selecciona el torneo y la edición de la que desea obtener información.	
Resultados Esperados: Se visualiza en una vista con varios datos organizados que informan sobre la estructura y los resultados del evento seleccionado.	

Evaluación de la prueba: Satisfactoria.

Tabla 24: Caso de prueba para Generar principales líderes

Caso de Prueba de Aceptación	
Número : 12	Historia de Usuario: 12
Nombre: Generar principales líderes	
Condiciones de Ejecución: a)-Debe existir al menos un jugador con estadísticas registradas en el sistema.	
Entradas/Pasos de ejecución: - El usuario selecciona el torneo y la edición que desea obtener información mediante la tabla.	
Resultados Esperados: Se visualiza en una vista con los datos organizados que permiten identificar los líderes por cada departamento definido. Estos departamentos son: líder goleador, líder en asistencias, jugador con más tarjetas amarillas y rojas.	
Evaluación de la prueba: Satisfactoria.	

3.4. Resultados de las pruebas

Hasta el momento se han desarrollado un total de 45 casos de pruebas de aceptación. Estas pruebas fueron realizadas de forma organizada, por cada iteración definida. Como resultado se detectó que los errores encontrados pueden clasificarse según su tipo:

- Para una cantidad de 20 No Conformidades se revelan errores de Presentación porque no se visualiza correctamente toda la información que se desea mostrar.

- Para las siguientes 10 No Conformidades los errores fueron de Redacción porque se localizaron varios errores ortográficos siendo estos generalmente de acentuación.
- Para las restantes No Conformidades el error detectado fue de Ejecución, durante el despliegue del sistema fueron detectadas diferentes fallas, siendo estas ocasionadas porque el sistema no realizaba todos los procedimientos definidos.

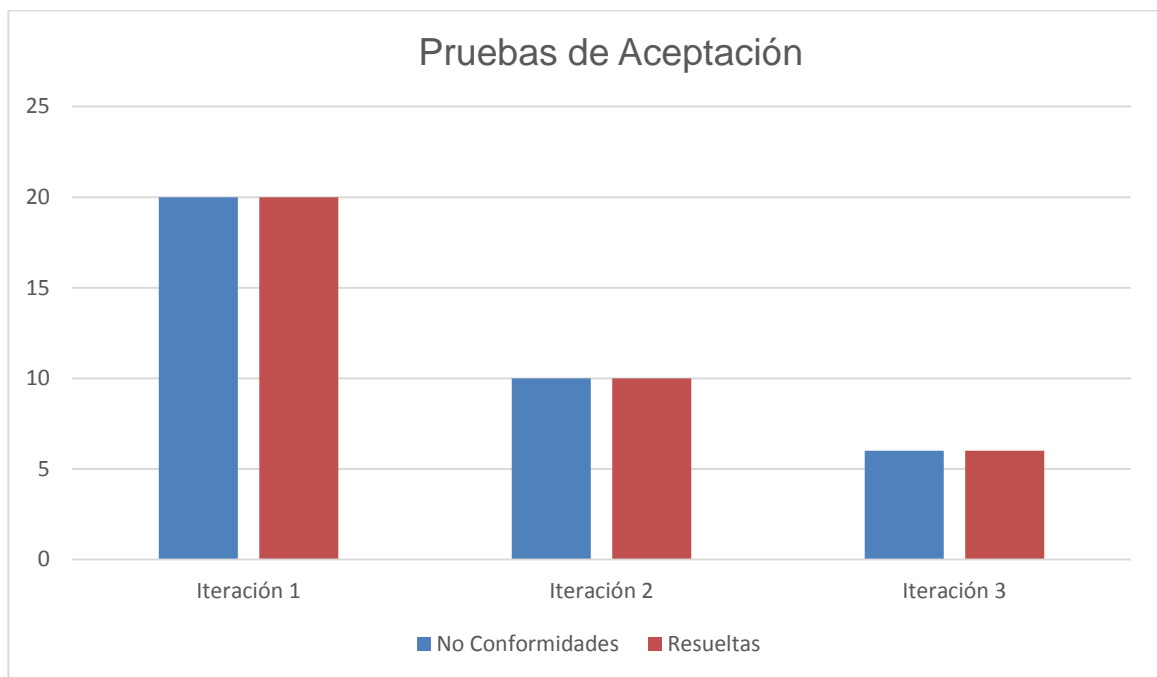


Figura 10: Resultados de las Pruebas de Aceptación

3.5. Conclusiones del capítulo

En este capítulo se especificó el proceso de implementación del sistema a partir del desglose de las HU en tareas de ingeniería, lo que permitió especificar los procedimientos necesarios para dar cumplimiento a cada HU. Además, se definieron y aplicaron las pruebas de aceptación a las funcionalidades de los módulos desarrollados. Estas pruebas permitieron detectar varias fallas en el sistema y corregirlas, lo que posibilitó mejorar la operabilidad del mismo.

Conclusiones generales

El actual trabajo se centró en el desarrollo del “Sistema para el análisis estadístico de información para estudiantes y profesores especialistas de fútbol del departamento de deportes de la Universidad de las Ciencias Informáticas”, con la finalidad de facilitar la toma de decisiones referente a movimientos estratégicos dentro del fútbol por parte de los profesores y especialistas del departamento de deportes de la UCI a partir de la información recolectada en los diferentes eventos que se celebran en este centro.

Para el desarrollo del sistema se efectuaron las siguientes actividades:

- El estudio teórico y práctico permitió centrar las bases para el desarrollo de la propuesta de solución.
- Se desarrolló una aplicación para el departamento de deportes de la UCI, la cual permite apoyar la toma de decisiones basada en el análisis estadístico de los jugadores de fútbol.
- Se desarrollaron las pruebas de aceptación lo que permitió comprobar la correcta implementación de las funcionalidades trazadas para el sistema SISFUT.

Como resultado de la investigación se logra el desarrollo de un software mediante el cual es posible gestionar la información respecto a las acciones que realizan los jugadores de fútbol en los partidos que estos sean partícipes. Esta herramienta es capaz de aplicar razonamiento basado en casos para estimar los jugadores ideales que deben representar a un equipo en su próxima jornada deportiva.

RECOMENDACIONES

A partir del resultado obtenido con el desarrollo de la presente investigación se recomienda la construcción de un módulo estadístico donde sea posible dar respuesta a las siguientes solicitudes:

- Los jugadores derechos/izquierdos que más goles anotan en un torneo.
- Como se comporta la media de goles por torneo.
- Los jugadores que más años llevan en un equipo.
- El jugador por el que más pasa el balón en un equipo.
- Por ciento de efectividad de pases de un jugador.
- Cantidad de disparos realizados por un jugador en un partido.
- Equipos que más faltas cometen en un torneo.
- Equipos líderes en asistencias, goles, amarillas, rojas.

REFERENCIAS BIBLIOGRÁFICAS

- Aamodt, Agnar y Plaza, Enric. 1994.** Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. 1994. Vol. 7, 1, págs. 39-52.
- Álvarez Enq, Boris Luis y Rojas Güemes, Marlon. 2009.** *Software Educativo como soporte tecnológico del aprendizaje técnico-táctico del Fútbol para los estudiantes de la Universidad de las Ciencias Informáticas.* La Habana : s.n., 2009.
- Bello Pérez, Rafael Esteban. 2001.** *Tomando decisiones basadas en el conocimiento.* s.l. : Universidad Cooperativa de Colombia, 2001. 978-958-95568-2-5.
- Bello Pérez, Rafael Esteban, y otros. 2002.** *Aplicaciones de la Inteligencia artificial (. México : La Noche, 2002. 970-27-0177-5.*
- Borrego del Pino, Silvia. 2008.** *Estadística Descriptiva e Inferencial.* Granada : s.n., 2008. 1988-6047.
- Brodie, David y Thornhill, John. 1983.** *Microcomputing in Sport and Physical Education.* s.l. : A & C Black Publishers Ltd, 1983. 978-0860191063.
- Casillas Santillán, Luis Alberto, Gilbert Ginestà, Marc y Pérez Mora, Óscar. 2008.** Universidad Oberta de Catalunya OpenCourseWare. *Universidad Oberta de Catalunya OpenCourseWare.* [En línea] 06 de 05 de 2008. [Citado el: 15 de 2 de 2016.] http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02151.pdf.
- cms. 2008.** cms. [En línea] 27 de 4 de 2008. [Citado el: 30 de 4 de 2016.] <https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>.
- Cordero, J. 2010.** *Metodologías Ágiles "Proceso Unificado Ágil (AUP)".* 2010.
- Crámer, Harald. 1999.** *Mathematical Methods of Statistics.* s.l. : Princeton University Press, 1999. 978-0691005478.
- DATE, C. J. 2003.** *Introducción a los sistemas de bases de datos.* La Habana : Félix Varela, 2003.
- Donnelly, Joseph E. 1987.** *Using MicroComputers in Physical Education and the Sport Sciences.* s.l. : Human Kinetics Publishers, 1987. 0873220838 .
- EGUILUZ, Javier. 2009.** Introducción a CSS. LibrosWeb. Creative Commons. [En línea] 2009. [Citado el: 14 de Mayo de 2015.] <http://www.librosweb.es/css/index.html>.
- Eldritch, Mauro. 2015.** MDN Mozilla Developer Network. [En línea] 26 de Abril de 2015. [Citado el: 19 de Mayo de 2015.] <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Introducci%C3%B3n>.
- Escribano, G. 2002.** *Introducción a Extreme Programming.* 2002.
- Fernández, Dra. Josefa Marín. 2010-2011.** *Manual de prácticas con MINITAB 15.* Universidad de Murcia : Departamento de Estadística e Investigación Operativa, 2010-2011.
- Figuroa, Roberth G., Solís, Camilo J. y Cabrera, Armando A. 2008.** adonisnet. *adonisnet.* [En línea] 2008. [Citado el: 10 de 5 de 2016.] <http://adonisnet.wordpress.com/2008/06/18/metodologias-tradicionales-vs-metodologias-agiles..>
- Fundation, jQuery. 2016.** jQuery. *jQuery.com.* [En línea] 2016. <https://jquery.com/>.
- FutStats. 2013.** Fútbol en positivo. *Fútbol en positivo.* [En línea] 2013. [Citado el: 5 de 5 de 2016.] <http://futbolenpositivo.com/index.php/futstats-hd-controla-a-tu-equipo-dese-tu-ipad-o-iphone/>.
- GOMEZ, Daniel. 2010.** *Programación Extrema.* 2010.
- González, Gregorio Morales. 2009.** *Plan de acción para potencial el fútbol en la facultad # 9 en la Universidad de las Ciencias informáticas.* La Habana : s.n., 2009.
- González, Y. 2012.** Patrón Modelo-Vista-Controlador. s.l. : Revista Telemática, 2012.

- Group, Netbeans.** 2016. NetBeans IDE - Overview. *Netbeans.org*. [En línea] 2016. <https://netbeans.org/features/index.html>.
- Handball Match Analysis: Computerized Notation System.** Skarbalius, Antanas y Krusinskiene, Radvile. 2003. 1, s.l. : Int. J. Comp. Sci. Sport, 2003, Vol. 2.
- InnovaFutsal.** 2015. InnovaFutsal. *InnovaFutsal*. [En línea] Innova B2b Solutions, 2015. [Citado el: 3 de 4 de 2016.] <http://www.innovafutsal.com/>.
- Jacobson, Ivar y Rumbaugh, James.** 2000. *El Proceso Unificado de Desarrollo de Software*. Madrid : Madrid, 2000. 84-7829-036-2.
- JetBrains.** 2016. PhpStorm IDE :: JetBrains PhpStorm. *JetBrains*. [En línea] 2016. <https://www.jetbrains.com/phpstorm/>.
- JOSÉ H. CANÓS, P. L., M^a CARMEN PENADÉS.** 2002. *Metodologías Ágiles en el Desarrollo de Software* . 2002.
- Joskowicz, José.** 2008. *Reglas y prácticas en eXtreme Programming*. s.l. : Universidad de Vigo, 2008.
- LARMAN, Craig.** 2003. *UML y Patrones Introducción al análisis y diseño orientado a objeto*. s.l. : Pearson, 2003.
- Letelier, Patricio y Penades, M^a Carmen.** 2013. *Métodologías ágiles para el desarrollo de software:eXtreme . Programming (XP)*. 2013.
- Loaiza, Douglas Alfredo.** 2010. *pnfiingenieriadesoftwaregrupocuatro*. *pnfiingenieriadesoftwaregrupocuatro*. [En línea] 2010. [Citado el: 12 de 3 de 2016.] <http://pnfiingenieriadesoftwaregrupocuatro.blogspot.com/2012/07/bienvenidos-al-blog.html>.
- Martínez, A. M. R.** 2003. *Guía a Rational Unified Process*. 2003.
- MASTERMAGAZINE.** 2014. Mastermagazine. [En línea] 2014. [Citado el: 19 de Mayo de 2015.] <http://www.mastermagazine.info/termino/3866.php>.
- Meriño, C.** 2008. *Programación en PHP5. Nivel Básico*. Septiembre de 2008.
- MICROSYSTEMS.** 2016. NetBeans IDE 8.0.2 Release Information. *NetBeans.org*. [En línea] 2016. <https://netbeans.org/community/releases/80/>.
- Morán, Victor D. Pinilla.** 2009. *Análisis estadístico de datos muestrales*. 2009.
- Munar, Joaquín Alegre Martín y Magdalena Cladera.** 2002. *INTRODUCCIÓN A LA ESTADÍSTICA DESCRIPTIVA PARA ECONOMISTAS*. Palma : s.n., 2002.
- OPENSUSE.** 2014. opensuse. *opensuse*. [En línea] 2014. [Citado el: 19 de Mayo de 2015.] <https://es.opensuse.org/Apache>.
- Paganelli, Cesar.** 2000. http://www.futsala.com.ar/sitioviejo/historia_del_futsal_inf.htm. http://www.futsala.com.ar/sitioviejo/historia_del_futsal_inf.htm. [En línea] 2000. [Citado el: 21 de 02 de 2016.]
- Pavan, Bárbara.** 2013. ENTENDIENDO HTML5: GUÍA PARA PRINCIPIANTES. [En línea] 2013. [Citado el: 20 de 2 de 2016.] <http://bitelia.com/2013/05/entendiendo-html5-guia-para-principiantes>.
- Payrol Trillo, Loreta y González Torres, María de los Ángeles.** 2008. *Sistema de Información Estadístico Complementario de Salud: Módulo Infección Intrahospitalaria*. La Habana : s.n., 2008.
- Pressman, R. S.** 2003. *Ingeniería del Software: una tecnología estratificada. Ingeniería del Software. Un enfoque práctico*. 2003.
- Rodríguez, Txema.** 2012. Bootstrap. *Genbetadev.com*. [En línea] 2012. <http://www.genbetadev.com/frameworks/bootstrap>.
- Rodriguez-Perez, JJ.** 2008. *SISTEMA DE RECUPERACIÓN DE INFORMACIÓN DE MERCADOTECNIA PROVENIENTE DE INTERNET PARA ALBET S.A*. 2008.
- Ruiz Muñoz, David.** 2004. *Manual de Estadística*. s.l. : Eumed, 2004. 84-688-6153-7.

- RUSSELL, S. J.: PÍORVIG, P. 2004.** *INTELIGENCIA ARTIFICIAL. UN ENFOQUE MODERNO*. Madrid : s.n., 2004.
- Schenone, Hernán Marcelo. 2004.** *Diseño de una Metodología Ágil de Desarrollo de Software*. 2004.
- Schwaber, K. 1997.** *Scrum development process. Business Object Design and Implementation*. 1997.
- Sharp, B. 1996.** The use of computers in sport science. *British Journal of Education Lechochnology*. 1996, Vol. 27, 1.
- Suterland, Ph. D. 2007.** *Distributed Scrum: Agile project management with outsourced development teams*. 2007.
- VALDÉS, Damián Pérez. 2007.** Maestros del Web. [En línea] 3 de Julio de 2007. [Citado el: 19 de Mayo de 2015.] <http://www.maestrosdelweb.com/que-es-javascript/>.
- VisualParadigm.com. 2016.** What is Visual Paradigm? *Visual-paradigm.com*. [En línea] Visual Paradigm International, 2016. <https://www.visual-paradigm.com/features/>.
- Zend. 2015.** The PHP IDE for Smarter Development - boost productivity. *Zend.com*. [En línea] 2015. <http://www.zend.com/en/products/studio>.

CAPÍTULO 4. ANEXOS

Anexo I: Historias de Usuarios.

Tabla 25: HU7 Adicionar edición.

Historia de Usuario	
Número: 7	Nombre: Adicionar edición
Usuario: Administrador	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Media
Puntos Estimados: 0.5	Iteración asignada: 1
Programador responsable: Yalbert González Antúnez	
<p>Descripción: Permite al administrador adicionar una edición. Datos para adicionar: -Número de la edición -Temporada -Sistema de competición -Fecha de la edición</p>	
<p>Observaciones: Para que el administrador pueda adicionar una edición debe de haber creado algún torneo para incluir la edición nueva.</p>	

Interfaz:

Adicionar nueva edición

Torneos > Adicionar edición

Número de la edición
#

Temporada
Ente organizador

Sistema de la competición
Ente organizador

Fecha de la edición
dd/mm/aaaa

Insertar Cancel

Tabla 26: HU5 Adicionar equipo.

Historia de Usuario	
Número: 5	Nombre: Adicionar equipo
Usuario: Administrador	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Media
Puntos Estimados: 0.5	Iteración asignada: 1
Programador responsable: Yalbert González Antúnez	

Descripción:

Permite al administrador adicionar un equipo.

Datos para adicionar:

- Nombre del equipo
- Ente
- Región
- Fecha de fundación
- Adjuntar camiseta
- Adjuntar escudo
- Adjuntar foto

Observaciones: Para que el administrador pueda crear un equipo debe adicionarse una edición anteriormente.

Interfaz:

The screenshot shows a web interface for adding a new team. At the top, it says 'Adicionar nuevo torneo'. Below that is a breadcrumb trail: 'Equipos ediciones > Nuevo equipo'. A green button labeled 'Nuevo equipo' is visible. The form includes several fields: 'Nombre equipo' (with a dark input field), 'Ente' (with a dropdown menu), 'Región' (with a dropdown menu), and 'Fecha de fundación' (with a date picker). There are three sections for attachments: 'Adjuntar Camiseta', 'Adjuntar Escudo', and 'Adjuntar Foto', each with an 'Examinar...' button and the text 'No se ha seleccionado ningún archivo.' At the bottom right, there are two buttons: 'Insertar' (green) and 'Cancel' (grey).

Tabla 27: HU29 Eliminar equipo

Historia de Usuario	
Número: 3	Nombre: Eliminar equipo
Usuario: Administrador	
Prioridad en el negocio: Baja	Riesgo de desarrollo: Baja
Puntos Estimados: 0.1	Iteración asignada: 1
Programador responsable: Yalbert González Antúnez	
Descripción: Permite al administrador eliminar un equipo.	
Observaciones: Permite que el administrador pueda eliminar un equipo que haya sido creado.	
Interfaz:	

Tabla 28: HU30 Eliminar edición

Historia de Usuario	
Número: 2	Nombre: Eliminar edición
Usuario: Administrador	
Prioridad en el negocio: Baja	Riesgo de desarrollo: Baja
Puntos Estimados: 0.1	Iteración asignada: 1

Programador responsable: Yalbert González Antúnez
Descripción: Permite al administrador eliminar una edición.
Observaciones: Permite que el administrador pueda eliminar una edición que haya sido creada.
Interfaz:

Tabla 29: HU14 Generar evento competitivo

Historia de Usuario	
Número: 11	Nombre: Generar evento competitivo
Usuario: Administrador	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Alta
Puntos Estimados: 1.5	Iteración asignada: 3
Programador responsable: Yalbert González Antúnez	
Descripción: Permite al administrador mostrarles a los diferentes usuarios una tabla con los resultados de todos los partidos realizados en ese evento competitivo.	
Observaciones:	

Interfaz:

Tabla 30: *HU38 Listar ediciones*

Historia de Usuario	
Número: 2	Nombre: Listar ediciones
Usuario: Administrador	
Prioridad en el negocio: Baja	Riesgo de desarrollo: Baja
Puntos Estimados: 0.1	Iteración asignada: 1
Programador responsable: Yalbert González Antúnez	
Descripción: Permite al administrador listar todas las ediciones creadas.	
Observaciones: Para poder listar las ediciones tienen que haber sido creadas antes por el administrador.	
Interfaz:	

Tabla 31: *HU39 Listar equipos.*

Historia de Usuario	
Número: 3	Nombre: Listar equipos
Usuario: Administrador	

Prioridad en el negocio: Baja	Riesgo de desarrollo: Baja
Puntos Estimados: 0.1	Iteración asignada: 1
Programador responsable: Yalbert González Antúnez	
Descripción: Permite al administrador listar todos los equipos creados.	
Observaciones:	
Interfaz:	

Tabla 32: HU40 Listar jornadas.

Historia de Usuario	
Número: 4	Nombre: Listar jornadas
Usuario: Administrador	
Prioridad en el negocio: Baja	Riesgo de desarrollo: Baja
Puntos Estimados: 0.1	Iteración asignada: 1
Programador responsable: Yalbert González Antúnez	
Descripción: Permite al administrador listar todas las jornadas creadas.	

Observaciones:
Interfaz:

Tabla 33: HU41 Listar calendario.

Historia de Usuario	
Número: 5	Nombre: Listar calendario
Usuario: Administrador	
Prioridad en el negocio: Media	Riesgo de desarrollo: Baja
Puntos Estimados : 0.1	Iteración asignada: 2
Programador responsable: Yalbert González Antúnez	
Descripción:	
Observaciones: Para que el administrador pueda listar los calendarios tienen que haber sido creados antes.	
Interfaz:	

Tabla 34: HU42 Listar jugadores.

Historia de Usuario	
Número: 6	Nombre: Listar jugadores

Usuario: Administrador	
Prioridad en el negocio: Baja	Riesgo de desarrollo: Baja
Puntos Estimados: 0.1	Iteración asignada: 2
Programador responsable: Yalbert González Antúnez	
Descripción: Permite al administrador listar todos los jugadores creados.	
Observaciones: Permite al administrador listar todos los jugadores creados anteriormente	
Interfaz:	

Tabla 35: HU43 Listar contratos.

Historia de Usuario	
Número: 7	Nombre: Listar contratos
Usuario: Administrador	
Prioridad en el negocio: Baja	Riesgo de desarrollo: Baja
Puntos Estimados: 0.1	Iteración asignada: 2
Programador responsable: Yalbert González Antúnez	

<p>Descripción: Permite al administrador listar todos los contratos creados.</p>
<p>Observaciones:</p>
<p>Interfaz:</p>

Tabla 36: HU44 Listar árbitros.

Historia de Usuario	
Número: 8	Nombre: Listar árbitros
Usuario: Administrador	
Prioridad en el negocio: Baja	Riesgo de desarrollo: Baja
Puntos Estimados: 0.1	Iteración asignada: 2
Programador responsable: Yalbert González Antúnez	
<p>Descripción: Permite al administrador listar todos los árbitros creados.</p>	
<p>Observaciones:</p>	
<p>Interfaz:</p>	

Tabla 37: HU44 Listar directores técnicos.

Historia de Usuario	
Número: 9	Nombre: Listar directores técnicos.
Usuario: Administrador	
Prioridad en el negocio: Baja	Riesgo de desarrollo: Baja
Puntos Estimados: 0.1	Iteración asignada: 3
Programador responsable: Yalbert González Antúnez	
Descripción: Permite al administrador listar todos los directores técnicos creados.	
Observaciones:	
Interfaz:	

Tabla 38: HU45 Listar noticias.

Historia de Usuario	
Número: 10	Nombre: Listar noticias.
Usuario: Administrador	
Prioridad en el negocio: Baja	Riesgo de desarrollo: Baja
Puntos Estimados: 0.1	Iteración asignada: 3
Programador responsable: Yalbert González Antúnez	

Descripción: Permite al administrador listar todas las noticias.
Observaciones:
Interfaz:


Tabla 39: *HU12 Generar principales líderes.*

Historia de Usuario	
Número: 12	Nombre: Generar principales líderes.
Usuario: Administrador	
Prioridad en el negocio: Alta	Riesgo de desarrollo: Alta
Puntos Estimados: 1.5	Iteración asignada: 4
Programador responsable: Yalbert González Antúnez	
Descripción: Permite al administrador mostrarles a los diferentes usuarios una tabla con los resultados de los líderes en cada departamento, como por ejemplo goles, asistencias entre otros.	
Observaciones: Para que el administrador pueda mostrar esta información a los diferentes usuarios debe haberse jugado al menos un juego.	

Interfaz:

LIDERES EN LA COPA


GOLEADORES
ASISTENCIAS
AMARILLAS
ROJAS



Nombre: Glez

Equipo: Facultad 5

3



Nombre: Perez

Equipo: FICl

2

Anexo II: Tarjetas CRC

Tabla 40: Tarjeta CRC insertarJugadores.

Tarjeta CRC	
Clase: insertarJugadores	
Responsabilidades:	Colaboradores:

<p>1-Adiciona un nuevo jugador.</p> <p>2-Busca por el id de jugador cualquier jugador adicionado.</p> <p>3-Lista todos los jugadores adicionados.</p> <p>4-Modifica un jugador adicionado.</p> <p>5-Elimina cualquier jugador adicionado.</p>	<p>- insertarContratos</p>
---	----------------------------

Tabla 41: Tarjeta CRC insertarContratoJugador.

Tarjeta CRC	
Clase: insertarContratoJugador	
Responsabilidades:	Colaboradores:
1-Adiciona un contrato a ese jugador adicionado. 2-Busca los contratos por el id del mismo. 3-Lista todos los contratos hechos a los jugadores adicionados. 4-Modifica un contrato a cualquier jugador adicionado. 5-Elimina el contrato de un jugador adicionado.	- insertarContratos

Tabla 42: Tarjeta CRC insertarDT.

Tarjeta CRC	
Clase: insertarDT	
Responsabilidades:	Colaboradores:

<p>1-Adiciona un DT a un equipo.</p> <p>2-Busca los DT por el id del mismo.</p> <p>3-Lista todos los DT adicionados.</p> <p>4-Modifica un DT de cualquier equipo adicionado.</p> <p>5-Elimina el DT de un equipo adicionado.</p>	<p>- insertarTorneos</p>
--	--------------------------

Anexo III: Tareas de Ingeniería

Tabla 43: Tarea de ingeniería # 7

Tarea	
Número de tarea: 7	Número de HU: 7,21,32,41
Nombre de la tarea: Gestionar contratos	
Programador responsable: Yalbert González Antúnez	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 8 de febrero del 2016	Fecha de fin: 15 de febrero del 2016
Descripción: Cuando se termina la tarea le permite al administrador poder insertar un contrato, eliminar un contrato creado anteriormente, modificar cualquier contrato creado o listar todos los contratos que hayan sido creados.	

Tabla 44: Tarea de ingeniería # 8

Tarea	
Número de tarea: 8	Número de HU: 8,22,33,42
Nombre de la tarea: Gestionar árbitros	
Programador responsable: Yalbert González Antúnez	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 17 de febrero del 2016	Fecha de fin: 24 de febrero del 2016
Descripción: Cuando se termina la tarea le permite al administrador poder insertar un árbitro, eliminar un árbitro creado anteriormente, modificar cualquier árbitro creado o listar todos los árbitros que hayan sido creados.	

Tabla 45: Tarea de ingeniería # 9

Tarea	
Número de tarea: 9	Número de HU: 9,23,34,43
Nombre de la tarea: Gestionar directores	
Programador responsable: Yalbert González Antúnez	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 26 de febrero del 2016	Fecha de fin: 4 de marzo del 2016
Descripción: Cuando se termina la tarea le permite al administrador poder insertar un director, eliminar un director creado anteriormente, modificar cualquier director creado o listar todos los directores que hayan sido creados.	

Tabla 46: Tarea de ingeniería # 10

Tarea	
Número de tarea: 10	Número de HU: 10,24,35,44
Nombre de la tarea: Gestionar noticias	
Programador responsable: Yalbert González Antúnez	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 5 de marzo del 2016	Fecha de fin: 12 de marzo del 2016
Descripción: Cuando se termina la tarea le permite al administrador poder insertar una noticia, eliminar una noticia creada anteriormente, modificar cualquier noticia creada o listar todas las noticias que hayan sido creadas.	

Tabla 47: Tarea de ingeniería # 11

Tarea	
Número de tarea: 11	Número de HU: 11
Nombre de la tarea: Generar evento competitivo	
Programador responsable: Yalbert González Antúnez	
Tipo de tarea: Desarrollo	Puntos estimados: 1.2
Fecha de inicio: 14 de marzo del 2016	Fecha de fin: 21 de marzo del 2016
Descripción: El administrador después de terminar la tarea puede mostrar en una tabla algunos datos como nombre del torneo, jornadas creadas, equipos que juegan en esa jornada, entre otros.	

Tabla 48: Tarea de ingeniería # 12

Tarea	
Número de tarea: 12	Número de HU: 12
Nombre de la tarea: Generar principales líderes	
Programador responsable: Yalbert González Antúnez	
Tipo de tarea: Desarrollo	Puntos estimados: 1.5
Fecha de inicio: 4 de abril del 2016	Fecha de fin: 21 de abril del 2016
Descripción: El administrador después de terminar la tarea puede mostrar en una tabla los principales líderes en cada uno de los departamentos como líder goleador de ese torneo, líder en asistencias en ese torneo, jugador con más amarillas, jugador con más rojas, entre otras cosas.	

Tabla 49:Tarea de ingeniería # 13

Tarea	
Número de tarea: 13	Número de HU: 13
Nombre de la tarea: Generar las estadísticas finales de cada partido	
Programador responsable: Yalbert González Antúnez	
Tipo de tarea: Desarrollo	Puntos estimados: 1.2
Fecha de inicio: 25 de abril del 2016	Fecha de fin: 13 de mayo del 2016
Descripción: El administrador después de terminar la tarea puede mostrar en una tabla un resumen general de todo lo ocurrido en el partido, como por ejemplo, cantidad de tiros a puerta, tiros entre los 3 palos, los goles, asistencia, amarillas, cantidad de faltas, los jugadores que fueron titulares entre otros datos.	

Tabla 50:Tarea de ingeniería # 14

Tarea	
Número de tarea: 14	Número de HU: 14
Nombre de la tarea: Generar los 5 mejores jugadores de un equipo.	
Programador responsable: Yalbert González Antúnez	
Tipo de tarea: Desarrollo	Puntos estimados: 1.2
Fecha de inicio: 25 de abril del 2016	Fecha de fin: 13 de mayo del 2016
Descripción: El administrador después de terminar la tarea puede mostrar en una tabla los 5 mejores jugadores que han jugado en ese torneo y así el entrenador puede elegir los jugadores para la próxima fecha.	

Anexo IV: Modelo de datos

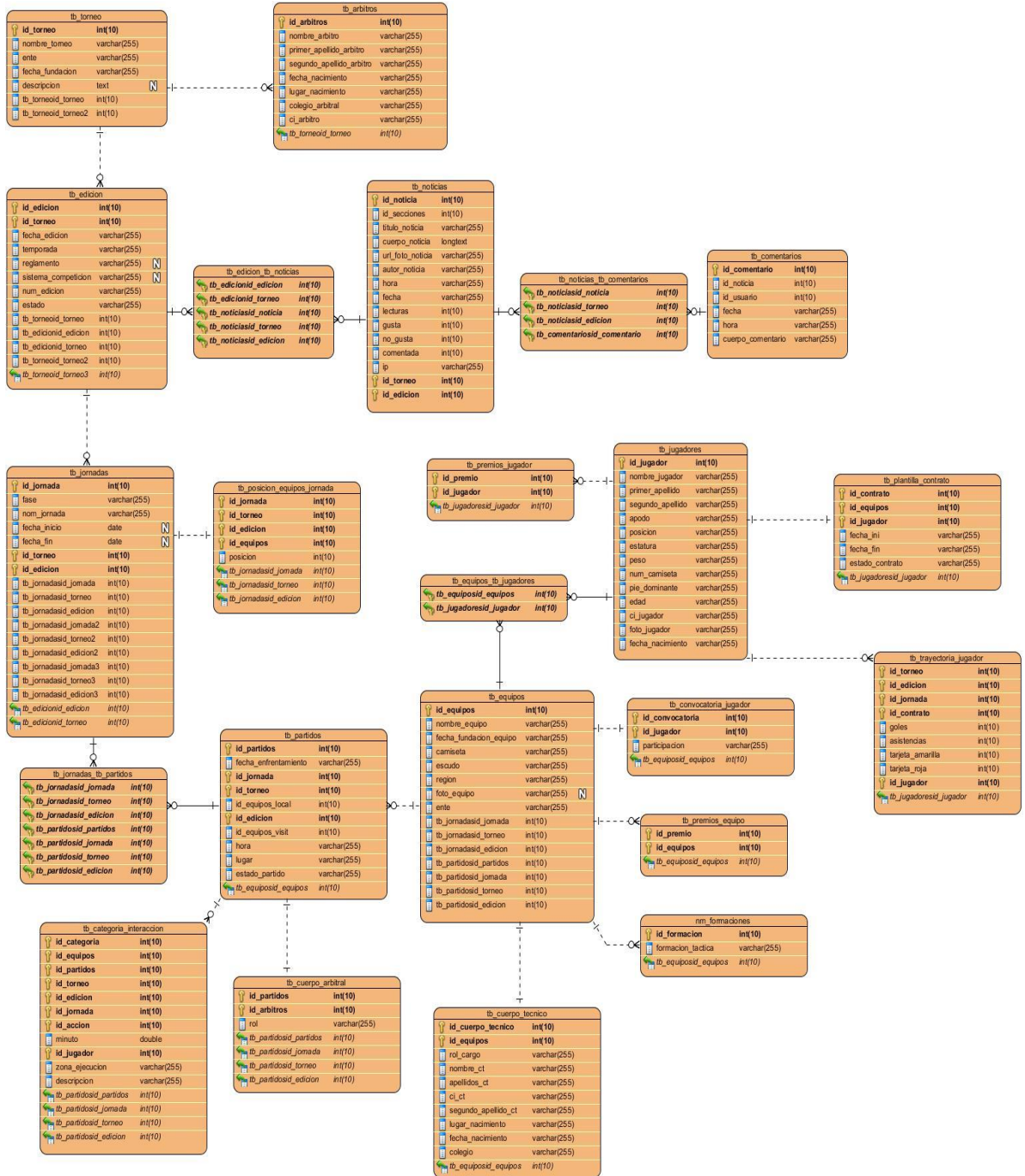


Figura 11: Modelo de Datos.