

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

**“SISTEMA DE GESTIÓN Y SEGUIMIENTO TÉCNICO PARA
ESCUELAS DEPORTIVAS DE EL ALTO”**

CASO: ESCUELAS DEPORTIVAS DE FUTBOL FUNDACIÓN ADEFUSA

Para optar al título de Licenciatura en Ingeniería de Sistemas

MENCIÓN: GESTIÓN Y PRODUCCIÓN

Postulante: Daniel Florencio Paye Aruquipa.

Tutor Metodológico: Ing. Marisol Arguedas Balladares.

Tutor Especialista: Ing. Sergio Ramiro Rojas Saire.

Tutor Revisor: Ing. Wendy Yomar Sarmiento Martínez

**EL ALTO - BOLIVIA
2020**

Dedicatoria

A mi esposa, mis hijas, mis padres y mi familia, por el apoyo incondicional para lograr este objetivo, que sin ellos no hubiera sido posible en alcanzar este sueño añorado por mucho tiempo.

A mis tutores, por guiarme en la culminación de este trabajo de grado, que no ha sido fácil, pero se logró con el objetivo trazado.

Agradecimientos

A Dios, ante todo por darme la oportunidad conocer este mundo tan maravilloso, por darme la fuerza de poder seguir encaminando por los senderos del conocimiento, la perseverancia poder cumplir mis sueños y objetivos que uno se traza en la vida.

A mi Tutor Metodológico, Ing. Marisol Arguedas Balladares por su paciencia, sabiduría, tiempo y motivación que me brindo desde el momento que empecé con este proyecto.

A mi Tutor Especialista, Ing. Sergio Ramiro Rojas Saire, por compartir sus conocimientos, su experiencia creando una línea de amistad y compromiso con el trabajo de grado llegando así a la culminación de este proyecto.

A mi Tutor Revisor, Ing. Wendy Yomar Sarmiento Martínez, por encaminar el proyecto para lograr los objetivos, que gracias a las observaciones y recomendaciones brindadas se logró realizar un buen trabajo.

A la Universidad Pública de El Alto y en especial a la Carrera Ingeniería de Sistemas, por acogerme en sus aulas por los años de estudio

Y por último a la Fundación ADEFUSA, quien me brindo el apoyo incondicional otorgando todas las facilidades para que este proyecto pueda ser realizada, sin lugar a duda el sistema será de gran apoyo a la institución.

RESUMEN

El presente proyecto fue desarrollado, a fin de automatizar procesos manuales para reducir tiempo en la planificación de entrenamiento y entrega de reportes de rendimiento de deportistas.

En el presente proyecto se desarrolló el “Sistema Gestión y Seguimiento Técnico para escuelas deportivas de la ciudad de El Alto, caso de específico, escuelas de Fútbol Fundación Adefusa, cumpliendo el objetivo del mismo, al brindar la información necesaria y relevante al personal de la institución.

Para el desarrollo de la Aplicación Web se aplicó la metodología UWE, para evaluar la calidad se utilizó la ISO 9126, la estimación de costo del software se lo hizo con COCOMO II basándonos en puntos de función.

Las herramientas de desarrollo que se usaron fueron: MariaDB 10.4.11 como gestor de base de datos, PHP 7.4.5 como lenguaje de programación, Apache 2.4.43 como servidor HTTP, Bootstrap 4 como, como Frontend y Backend el framework Yii2.

CONTENIDO

1. Marco Preliminar	1
1.1. Introducción	1
1.2. Antecedentes	3
1.2.1. Antecedente Institucional	3
1.2.2. Antecedente Nivel Nacional	4
1.2.3. Antecedente Nivel Internacional	4
1.3. Planteamiento del Problema	6
1.3.1. Problema General	7
1.3.2. Problemas Específicos	7
1.4. Objetivos	7
1.4.1. Objetivo General	7
1.4.2. Objetivos Específicos	8
1.5. Justificación	8
1.5.1. Justificación Técnica	8
1.5.2. Justificación Económica	9
1.5.3. Justificación Social	9
1.6. Metodología	10

1.6.1.	Ingeniería de Software	10
1.6.2.	Metodología de Desarrollo	10
1.6.3.	Proceso de Software	11
1.6.4.	Método de Pruebas de Sistema	11
1.6.5.	Métrica de Calidad de Software	12
1.7.	Herramientas	13
1.7.1.	Herramienta de para Modelado de Base de datos	13
1.7.2.	Herramienta de Desarrollo	13
1.7.3.	Herramienta de Gestor de Base de Datos	13
1.7.4.	Servidor de Aplicaciones Web	14
1.8.	Límites y Alcances	14
1.8.1.	Límites	14
1.8.2.	Alcances	14
1.9.	Aportes	15
Marco Teórico		16
2.1.	Introducción	16
2.2.	Administración Deportiva	16
2.2.1.	Funciones del Administrador Deportivo	16
2.3.	Gestión Deportiva	17
2.4.	Seguimiento Técnico	17

2.5.	Rendimiento Deportivo	18
2.5.1.	Planificación Deportiva	18
2.5.2.	Componentes de la Carga de Entrenamiento	19
2.5.3.	Condicionantes del Entrenamiento	21
2.6.	Componentes de la Planificación deportiva	23
2.6.1.	Estructura de Proceso de Entrenamiento	23
2.7.	Sistema	26
2.8.	Sistema de Gestión	27
2.9.	Ingeniería de software	27
2.9.1.	El proceso de software	27
2.9.2.	Aplicaciones Web	28
2.10.	Metodología UWE	29
2.10.1.	Modelo de Requerimientos	29
2.10.2.	Modelo de Contenido (conceptual) y modelo de usuario	30
2.10.3.	Modelo de Navegación	30
2.10.4.	Modelo de Presentación	31
2.11.	Ciclo de Vida del Software	31
2.11.1.	Modelo Iterativo e Incremental	32
2.12.	Orientada a Objetos	32

2.12.1.	Objeto	32
2.12.2.	Clase	33
2.12.3	Herencia	35
2.12.4.	Interfaz	36
2.13.	Diagramas UML	37
2.13.1.	Bloques básicos de construcción de UML	37
2.13.2.	Diagrama de Clases y Diagrama de Objetos	41
2.13.3.	Diagrama de Casos de Uso	44
2.13.4.	Diagrama de Secuencia y Diagrama de Colaboración	49
2.13.5.	Diagrama de Estados y Diagrama de Actividades	52
2.14.	Calidad de Software	54
2.14.1.	Normas de Calidad Internacionales (ISO)	55
2.14.2.	ISO 9126	56
2.14.3.	Modelos, Métricas e Instrumentos para Medir Calidad en Uso.	56
2.15.	Pruebas de Calidad de Software	59
2.15.1.	Tipos de Pruebas	59
2.16.	Estimación de Costos de Desarrollo de Software	59
2.16.1.	Modelo COCOMO.	59
Marco Aplicativo		66
3.1.	Análisis de la Situación Actual	66

3.2.	Organigrama Institucional	66
3.3.	Escuelas de Fútbol Adefusa	67
3.4.	Ciclo de Vida del Software	69
3.5.	Fases de la Metodología UWE.	69
3.5.1.	Captura, Análisis y Especificación de Requisitos.	69
3.5.2.	Diseño del Sistema.	72
3.5.3.	Codificación de software.	79
3.3.4.	Pruebas.	89
3.3.5.	Mantenimiento.	109
	Calidad y Seguridad	110
4.1.	Calidad	110
4.2.	Metodología de evaluación calidad	110
4.3.	Fases	110
4.3.1.	Usabilidad.	110
4.3.2.	Funcionalidad.	112
4.3.3.	Confiabilidad.	117
4.3.4.	Mantenibilidad	118
4.3.5.	Portabilidad.	119
4.4.	Seguridad de Software	120
4.4.1	Autenticación.	120

4.4.2. Seguimiento a las Acciones de los Usuarios	121
4.4.3. Encriptación	122
4.4.4. Seguridad en la base de datos	123
Análisis de Costos	125
4.5. Modelo Cocomo II	125
4.6. Calculo de Costo	126
Conclusiones y Recomendaciones	132
6.1. Conclusiones.	132
6.2. Recomendaciones.	132
Bibliografía	134

ÍNDICE DE FIGURAS

Figura 2- 1:Modelo de Planificación	19
Figura 2- 2:Carga de Entrenamiento	22
Figura 2- 3 Capas de la Ingeniería de Software	28
Figura 2- 4 Objetos Comunicándose	34
Figura 2- 5 Ejemplo Herencia.....	36
Figura 2- 6 Diagrama de Clases.....	42
Figura 2- 7 Diagrama de Casos de uso	45
Figura 2- 8 Caso de uso Nivel 2	46
Figura 2- 9 Diagrama de Caso de Uso Extendido	46
Figura 2- 10 Diagrama de Secuencia	49
Figura 2- 11 Diagrama de Colaboración.....	52
Figura 2- 12 Máquina de Estados	53
Figura 2- 13 Diagrama de Actividades	54
Figura 2- 14 Modelo de Calidad	62
Figura 3- 1 Organigrama Adefusa	67
Figura 3- 2 Organigrama Escuela Deportiva	68
Figura 3- 3 Diagrama de caso uso del negocio	73

Figura 3- 4 Registrara Escuela.....	74
Figura 3- 5 Registrar deportista.....	74
Figura 3- 6 Crear Expediente deportista.....	75
Figura 3- 7 Crear Plan de entrenamiento	76
Figura 3- 8 Diagrama de clases	77
Figura 3- 9 Navegación de sistema.....	78
Figura 3- 10 inicio de sesión	79
Figura 3- 11 pantalla de Bienvenida.....	80
Figura 3- 12 Control de Accesos	81
Figura 3- 13 Escuelas Deportivas	82
Figura 3- 14 Detalle Escuela Deportiva	83
Figura 3- 15 Cargo Personal	84
Figura 3- 16 Detalle Personal.....	85
Figura 3- 17 Información personal.....	85
Figura 3- 18 información Deportista	86
Figura 3- 19 Plan Anual de Entrenamiento.....	87
Figura 3- 20 Evaluación Deportista	88
Figura 3- 21 Valoración Deportista.....	89

Figura 4- 1 Validación	121
Figura 4- 2 Seguimiento Acciones del usuario	122
Figura 4- 3 Encriptación	123

ÍNDICE DE TABLAS

Tabla 2- 1 Tipos de Mesociclos.....	25
Tabla 2- 2 Elementos UML.....	38
Tabla 2- 3 Multiplicidad de Diagrama de Clases	42
Tabla 2- 4 Relaciones UML.....	43
Tabla 2- 5 Diagramas UML	47
Tabla 2- 6 Tipos de mensajes en diagramas de interacción.....	50
Tabla 2- 7 Esfuerzo de Personas-mes	62
Tabla 2- 8 Tabla de Coste de Software	63
Tabla 2- 9 Esfuerzo Aplicado en Líneas de Código.....	65
Tabla 3- 1 Técnica de recopilación de datos	69
Tabla 3- 2 Lista de Requerimientos Funcionales.....	70
Tabla 3- 3 Lista de Requerimientos no Funcionales.....	71
Tabla 3- 4 Prueba Caja Negra Registrar Escuela.....	89
Tabla 3- 5 Prueba Caja Negra Mantener Escuela	91
Tabla 3- 6 Prueba Caja Negra Mantener Personal.....	97
Tabla 3- 7 Prueba de Caja Negra Designar Escuela.....	102

Tabla 3- 8 Cuadro de Requerimientos de Software y Hardware.....	106
Tabla 3- 9 Actividades para la implementación	108
Tabla 4- 1 Tabla de Pregunta de Usabilidad	110
Tabla 4- 2 Tabla de funcionalidad	113
Tabla 4- 3 Factor de Ponderación	113
Tabla 4- 4 Puntos de Función	114
Tabla 4- 5 Módulos a evaluar Confiabilidad	117
Tabla 4- 6 Factores de Madurez	119
Tabla 4- 7 Funciones Mysql Yii2	124
Tabla 5- 1 Coeficiente del Modelo COCOMO II.....	126
Tabla 5- 2 Conversión de puntos de Función a KDLC.....	126
Tabla 5- 3 Costo de Implementación.....	129
Tabla 5- 4 Costo de Elaboración.....	130
Tabla 5- 5 Costo Total.....	131

Marco Preliminar

1.1. Introducción

El mundo deportivo se enfrenta constantes cambios, inmersos al constante cambio de la tecnología. También el deporte en la Ciudad de El Alto, se ha visto afectado, experimentando en los últimos años importantes cambios con relación a las motivaciones y actitudes de los ciudadanos con respecto a su práctica.

Efectivamente, se encuentra en una sociedad más moderna y dinámica, donde la elección de estilos de vida y una mayor preocupación por la salud y el disfrutar una vida lúcida, han propiciado un importante incremento en la práctica deportiva y en un concepto más complejo, supone la dificultad de poder determinar el comportamiento del deportista luego de una sesión de entrenamiento riguroso o leve.

La Fundación ADEFUSA (Academia de Deportes Fuerza y Salud), ha venido experimentando un moderado pero sostenido crecimiento de las escuelas de formación de deportistas en las diferentes disciplinas. Caso particular consiste disciplina de fútbol, ámbito en la que fue enmarcándose como uno de los pioneros en prestar sus servicios en la formación de deportistas. Además de incentivar a la creación de nuevas escuelas, coadyuvando con profesionales en el ámbito de la Gestión Deportiva, relacionada con las actividades que realiza un administrador, es decir, acciones encaminadas a planear, organizar, dirigir, ejecutar y controlar todo lo relacionado con el deporte.

Esta se lleva a cabo mediante programas y proyectos dedicados al aprovechamiento y proyección del deportista y se visualiza en escenarios donde la inclusión de la administración deportiva es fundamental. El gestor deportivo tiene que hacer frente a las situaciones adversas y situaciones imprevistas que quedan al margen de una ejecución habitual, de tal modo que su participación se refleje en la mejora del entorno deportivo. En torno a ello existen grupos diferenciados y un

número de deportistas en cada grupo, esto conlleva a la dificultad de controlar el trabajo realizado con los deportistas durante un tiempo determinado.

La problemática radica en que día a día existen mayor número de Escuelas Deportivas que forman parte de ADEFUSA y mayor cantidad de participantes, por lo que se hace más difícil la gestión y seguimiento técnico a las actividades del deportista dentro de una Escuela Deportiva en el Área de Fútbol, tales como inscripción del deportista, asignación de horarios, evaluación del deportista, generación de reportes y estadística, imposibilitando así la disponibilidad de la información confiable, pertinente y oportuna para la toma de decisiones.

La presente propuesta plantea desarrollar un **SISTEMA DE GESTIÓN Y SEGUIMIENTO TÉCNICO PARA ESCUELAS DEPORTIVAS DE EL ALTO en plataforma WEB**, que pueda optimizar los procesos de inscripción del deportista, asignación de horarios, evaluación del deportista, generación de reportes y estadística. Estos procesos son requeridos para coadyuvar en el trabajo del entrenador para realizar el seguimiento de los deportistas de fútbol en una gestión deportiva mediante programas y proyectos.

La propuesta se realizará bajo el método Descriptivo, la cual hará énfasis en la descripción de la forma de trabajo en la disciplina de fútbol respecto a la relación entre planificación, entrenador y deportista.

Para el desarrollo del proyecto se utilizará la metodología UWE (UML basada en la Ingeniería Web). Respecto al Modelado del Sistema se aplicará el Lenguaje Unificado, para el desarrollo de la programación el Framework PHP, más propiamente Yii Framework, es un potente entorno de trabajo mediante el método MVC (modelo, vista y controlador) facilitando el proceso de programación CRUD (Crear, Leer, actualizar y eliminar), esto va acompañada con un gestor de base de

datos MYSQL, siendo la herramienta más aplicada para aplicaciones de entornos WEB.

1.2. Antecedentes

1.2.1. Antecedente Institucional

ADEFUSA Bolivia, una entidad sin fines de lucro, cuya misión es apoyar a emprendimientos deportivos a nivel local, departamental y nacional. Realiza mediante programas y proyectos con visión a la promoción en niveles de recreación, formación y competencia en diferentes disciplinas deportivas. Caso concreto, en la disciplina de futbol, que viene trabajando desde los inicios de la conformación de esta fundación.

Misión. Promocionar de espacios de formación, intercambio y encuentro social, cultural y deportivo.

Visión. Constituir en una Fundación ejemplar y de referencia en la promoción de talentos deportivos en busca del alto nivel deportivo y competitivo.

Objetivo. Formar ciudadanos con principios y valores que permitan mejorar la calidad de vida individual, social y cultural que puedan responder a las exigencias del diario vivir.

La fundación cuenta con diferentes escuelas de futbol, situadas en diferentes zonas de la ciudad del El Alto, como ser en Zona Gran Poder, Villa Ingenio, Zenkata, Ciudad Satélite y otras zonas, donde vienen trabajando con la niñez y juventud alteña ejecutando proyectos deportivos como ICOTA (Identificación, Cooperación para el Talento Alteño), cuyo objetivo principal es la de trabajar con talentos mejorando la técnica y táctica del Jugador.

También cuenta con el proyecto Escuelas de Futbol, Escuelas de Vida, cuya colaboración de Ricerca E Cooperazione (fundación italiana) actualmente APEA, por una Educación Activa, son los brazos operativos y acción colaborativa con niños

y jóvenes de escasos recursos económicos y/o familias disgregadas, otorgando una oportunidad de vida mediante el fútbol.

1.2.2 Antecedente Nivel Nacional

[Ronald de Herrera, 2013], **El Programa “Electrón D.T.”**. Es un sistema para ayudar a través del internet la dirección técnica de los equipos la información procesada sistemáticamente sobre las habilidades y rendimiento de los jugadores a nivel profesional de un equipo o país.

Este software según el desarrollador, hace caculos complejos para ayudar a los técnicos en su evaluación de los jugadores, la que normalmente suele estar sólo bajo observación visual de las destrezas y a la revisión de las trayectorias.

También propone un conjunto de fórmulas matemáticas para medir las pruebas físicas de velocidad, reacción y agilidad, puntería, control del balón, cabeceo y la potencia en cada una de las piernas.

El objetivo del sistema es hace conocer con exactitud la estadística de las condiciones físicas que se encuentran cada jugador en cada momento.

1.2.3. Antecedente Nivel Internacional

Existen trabajos similares en otros países como se menciona en las siguientes líneas:

Daroqui Chafé, Julián Angel. 2016. Desarrollo de una aplicación web para la gestión de clubs deportivos y red social para sus miembros (Front-End).

Se centra, en la gestión de clubs deportivos, cuya característica destaca el control de contabilidad del club, creación y seguimiento de estadísticas a nivel individual y colectivo. Así como almacenar todas las localizaciones de las

instalaciones deportivas, coordinadas de geolocalización, para posteriormente ser asignadas a un partido.

Todo usuario registrado, tiene por defectos un rol invitado. Además, este rol, también se pueden asignar otros permisos al usuario dependiendo del estatus que este posea dentro del club, siendo los siguientes roles:

- **Administrador.** Control total sobre el club.
- **Entrenador.** Control total sobre el equipo que entrena.
- **Jugador.** Puede ser asignado a uno o varios equipos. El jugador posee un perfil con sus datos en los que aparecen estadísticas personales.
- **Padre/Madre.** Puede ser asignado a uno o varios equipos con la finalidad de estar informado de las últimas novedades del equipo al que ha sido asignado.

Los roles anteriores solo tienen valor dentro del club al que pertenece el usuario, por lo que si un miembro intenta acceder a otro club solo podrá ver la parte pública del mismo.

[Cebrian, D (n.d. 2010)], Gestión de Entrenamiento Deportivo, se ocupa completamente a la gestión deportiva y de los flujos de trabajo e información que relacionan a los sujetos entrenados con sus entrenadores. Lo más sobresaliente de este proyecto son:

- Administra usuarios, tales como administrador, entrenador y deportista.
- Los deportistas y entrenadores, son normalmente de deportes de velocidad, como atletismo, natación o cualquier deporte de competencia en velocidad.
- Crea sesiones de entrenamiento con fecha de inicio y final.
- Compara datos teóricos y los reportes obtenidos.
- Realiza una evaluación, entregando un reporte.
- Realiza estadísticas con tiempos anteriores.

[Consultores S.L. 2008], Gestión Deportiva, desarrollado para el mantenimiento y control de la información en clubes deportivos, especialmente en aquellos que manejan equipos de manera simultánea.

Lo más sobresaliente es lo siguiente:

- Administra usuario tales como administrador, entrenador y miembro del club.
- Los deportistas y entrenadores, son de diferentes deportes, como futbol, atletismo, voleibol, natación o cualquier deporte de competencia la cuales son registrados mediante un club.
- Crea sesiones de entrenamiento con fechas de inicio y final.
- Las sesiones de entrenamiento son vistos desde el punto de vista de un preparador físico, controlando el trabajo de frecuencia cardiaca, sudoración intensidad de trabajo y consumo energético.
- Compara datos teóricos y los reportes obtenidos.
- Realiza una evaluación, entregando un reporte.
- Realiza estadísticas con tiempos anteriores.

1.3. Planteamiento del Problema

La disponibilidad de información que posee la fundación ADEFUSA, sobre la gestión y seguimiento a sus escuelas deportivas se realiza de forma manual y en consecuencia se tiene información incompleta respecto a las escuelas creadas, personal designado, cantidad y grupos o categorías inscritas, avances y seguimiento en los planes de trabajo en el ámbito técnico. Por tal motivo se ve la necesidad de desarrollar un Sistema Gestión y Seguimiento Técnico para Escuelas Deportivas, que realice las tareas requeridas optimizando la correcta gestión y seguimiento de las escuelas de futbol de la Fundación ADEFUSA.

1.3.1. Problema General

ADEFUSA, viene tropezando con problemas de Gestión y Seguimiento Técnico de sus escuelas deportivas, debido a que las actividades deportivas como la inscripción de deportistas, asignación de escuelas y entrenadores, evaluaciones técnicas, son realizadas de forma manual y en consecuencia se tiene poca información de dichas actividades, dando a lugar el incumplimiento recurrente a la planificación realizada de los entrenadores y personal administrativo, conllevando a la mala formación y seguimiento del deportista.

1.3.2. Problemas Específicos

- La información respecto a los datos personales de los deportistas, el control de asistencia, sesiones avanzadas con el entrenador y evaluaciones realizadas durante el proceso de entrenamiento, son muy pocas y confusas, lo que conlleva a la improvisación de las sesiones de entrenamientos.
- El incumplimiento de la planificación de los entrenamientos de fútbol, hace que no se pueda desarrollar las actividades y objetivos en las sesiones de entrenamiento.
- El Actual método de seguimiento en el proceso de entrenamiento crea confusiones al momento de la valoración del deportista.
- No se tiene información con respecto a las actividades de las gestiones pasadas, lo que dificulta la toma de decisiones.

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar un Sistema Gestión y Seguimiento Técnico para Escuelas Deportivas en plataforma Web, para la fundación ADEFUSA, que permita el control

y seguimiento que optimicen los procesos de inscripción del deportista, asignación de horarios, evaluación, generación de reportes y estadística.

1.4.2. Objetivos Específicos

- Sistematizar, Organizar y clasificar las actividades según prioridad y necesidades de la fundación ADEFUSA.
- Aplicar métodos de seguimiento deportivo necesario en aspecto técnico en el tratamiento de la información.
- Automatizar el plan entrenamiento, contemplando ciclos y sesiones de entrenamiento.
- Automatizar la información generada por la realización de los requerimientos de la Fundación ADEFUSA.
- Realizar reportes de manera automatizada.
- Reducir errores en la generación y entrega de reportes.

1.5. Justificación

1.5.1. Justificación Técnica

Viendo la necesidad de aplicar el manejo de la información en las actividades de la Escuelas Deportivas que están bajo el convenio de la fundación ADEFUSA y satisfacer a los requerimientos que surgen de la misma.

El sistema Planteado será de plataforma Web, desarrollada con tecnologías libres utilizando la metodología UWE. El sistema podrá ser accedido con dispositivo que cuente con acceso a internet y en su navegador de internet, siendo el medio de enlace al sistema, por este motivo los requerimientos para los equipos con los cuales se podrá acceder a la plataforma tanto en hardware como en software son

los mínimos. Para la implementación del sistema se cuenta con un servicio de web Hosting de Corporativo, la cual mediante subdominios se creará un espacio para el sistema.

1.5.2. Justificación Económica

Los componentes del proyecto están ligados a la disponibilidad y uso de los recursos tecnológicos y los beneficios que presentan, como resultado del manejo adecuado. El sistema, ahorrara en gastos de impresión como en informes preliminares, o la transcripción de datos del deportista en medio físico que ésta conlleva en tiempo y dinero. Además, que el costo de material de escritorio se verá disminuida ya que la consulta de reportes y resultados serán vía web, quedando el interesado o la interesada en poder imprimirlo. El tiempo efectuado para tal sentido será en segundos optimizando así el tiempo para actividades pendientes dentro la institución.

1.5.3. Justificación Social

Sin duda, los estímulos sociales tienen un peso mayor que las limitantes económicas para ejecutar este proyecto. La entidad sin fines de lucro y de ayuda social busca establecer un bien común mediante el deporte, que si bien existe la pobreza no es un factor delimitante, sino un aspecto más de trabajo.

En tanto que un objetivo de la entidad constituye elevar el nivel de vida de los jugadores como activo de la entidad y el sistema planteado contribuye a tal objetivo, entregando como resultado el nivel y el proceso de formación de los jugadores como palestra de productos para promocionar a los deportistas resumiendo en una ficha técnica, mostrando en el portal web de la institución. Además, que levantará el nivel de profesionalismo de la entidad mostrando calidad en el trabajo realizado.

1.6. Metodología

1.6.1. Ingeniería de Software

La ingeniería de Software, se define como un “conjunto de etapas parcialmente ordenadas que persiguen un objetivo” que en este caso es la producción de un Software de Calidad Empresarial. Este proceso requiere una metodología y un lenguaje propio (ciclo de vida de software) [Lewis et. At 1994].

1.6.2. Metodología de Desarrollo

El desarrollo de software, no es una tarea fácil que debido a este problema ha surgido una alternativa llamadas Metodologías, que imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacer más predecible y eficiente. Existen diversas propuestas metodológicas en distintas dimensiones del desarrollo del software por mencionar algunas como la metodología RUP, SCRUM, XP, UWE, siendo UWE nuestra metodología a aplicar para este trabajo.

UWE. Es una metodología de la ingeniería de software para el domino web, que tienen como objetivo cubrir todo el ciclo de vida del desarrollo de aplicaciones web.

Fases o Etapas de la UWE.

- a) Captura, análisis y especificación de requisitos,** las características funcionales y no funcionales que debe cumplir la aplicación Web, mediante los diagramas de casos de uso.
- b) Diseño del Sistema,** es el diseño de los requisitos a cumplir del sistema.
 - Casos de uso.
 - Diagrama conceptual
 - Diagrama físico
 - Diagrama de clases

- Modelo navegacional
- Modelo de presentación

- c) Codificación de Software.**
- d) Pruebas**
- e) Instalación o implementación**
- f) Mantenimiento**

1.6.3. Proceso de Software

Se llama Procesos de Software, a las fases por las que pasa un proyecto de software desde que es concebido, hasta que está listo para usarse.

Típicamente, incluye las siguientes actividades: toma de requisitos, análisis, diseño, desarrollo, pruebas (validación, aseguramiento de la calidad), instalación (implantación), uso, mantenimiento y obsolescencia. El proyecto tiende a pasar iterativamente por estas fases, en lugar de hacerlo de forma lineal.

Para el estudio de caso, se ve por conveniente aplicar el **modelo incremental**, es el que permite que las fases de análisis, diseño, desarrollo y pruebas se retroalimentan continuamente, y que empiecen lo antes posible. Permitirá atender a posibles cambios a las necesidades del usuario, nuevas herramientas o componentes que los desarrolladores descubran y que faciliten el diseño y/o proporcionen nuevas funcionalidades.

1.6.4. Método de Pruebas de Sistema

Metodología FLOOT. La metodología de pruebas orientada a objetos para el ciclo de vida completo (Full Life-Cycle Object-Oriented Testing, FLOOT), es una colección de técnicas para verificar y validar software orientado a objetos. Indica una amplia variedad de técnicas que están disponibles en todos los aspectos de desarrollo de software.

1.6.5. Métrica de Calidad de Software

La calidad es un aspecto fundamental definiéndose a la calidad como “la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente” [R. S. Pressman (1992)].

ISO 9126-3. Métricas Internas de la Calidad del Producto del Software. ISO 9126, es un estándar internacional para la evaluación de la calidad de software. Este estándar está dividido en cuatro partes las cuales dirigen, respectivamente lo siguiente: modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso. El estándar califica la calidad de software en un conjunto estructurado de características de la siguiente manera:

- Funcionalidad, es el grado que satisface a las necesidades indicadas.
- Fiabilidad, cantidad de tiempo que el software está disponible para su uso.
- Usabilidad, grado en que el software hace óptimo el uso de los recursos del sistema.
- Eficiencia, grado en que el software hace óptimo el uso de los recursos del sistema.
- Mantenibilidad, facilidad con que una modificación puede ser realizada.
- Portabilidad, facilidad con que el software puede ser llevado en un entorno a otro.

1.7. Herramientas

Existen diversas herramientas de desarrollo que se adecuan al tipo de software a desarrollar, y se deben elegir la mejor opción para el trabajo en específico.

1.7.1. Herramienta de para Modelado de Base de datos

MySQL Workbench. Es una herramienta que permite modelar diagramas de entidad-relación para bases de datos **MySQL**. Se puede utilizar para diseñar el esquema de una base de datos nueva, documentar una ya existente o realizar una migración compleja.

1.7.2. Herramienta de Desarrollo

Php. Acrónimo recursivo de PHP: Hypertext Preprocessor, es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Yii. Es un framework PHP basado en componentes de alta performance para desarrollar aplicaciones Web de gran escala. El mismo permite la máxima reutilización en la programación web y puede acelerar el proceso de desarrollo.

1.7.3. Herramienta de Gestor de Base de Datos

MySQL. Es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales. Así, **MySQL** no es más que una aplicación que permite gestionar archivos llamados de bases de datos.

1.7.4. Servidor de Aplicaciones Web

HTTP Apache. Es un servidor web HTTP de código abierto, para plataformas Unix, Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio virtual.

1.8. Límites y Alcances

1.8.1. Límites

La presente propuesta, no contempla con todas las disciplinas deportivas, pues por ser el primer sistema desarrollado para la Entidad, será exclusivamente para el deporte de fútbol por albergar a mayor cantidad de deportistas, para luego expandir a disciplinas restantes.

1.8.2. Alcances

El sistema de información propuesta, tiene la facultad de realizar seguimiento y evaluar las actividades del deportista especialmente referentes al Fútbol, tomando como parámetro el rendimiento de deportivo conjuntamente con el plantel administrativo, para su mejor su mejor cumplimiento de las responsabilidades.

Para la implementación del sistema y para cumplir con los objetivos trazados se desarrollará los siguientes módulos:

- Módulo de registro de información del deportista, plantel de entrenadores y administrativo.
- Módulo de Designación de Horarios y Escuelas a deportistas
- Módulo de Designación de Horarios y Escuelas a entrenadores y personal administrativo
- Módulo de verificación de información registrada.
- Módulo de actualización de información generada.

- Módulo de evaluación del deportista.
- Módulo de registro del plan de entrenamiento por Escuelas y categorías de deportistas.
- Módulo de consultas o peticiones de información.
- Seguimiento de procesos del plan de trabajo de escuelas deportivas.
- Módulo de generación de reportes.

1.9. Aportes

Durante los años de formación en la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto y los conocimientos adquiridos en este tiempo, se realiza la sistematización del trabajo de la fundación ADEFUSA en sus escuelas deportivas, con un software que brinde apoyo en cuanto al área de tecnología de la información.

El sistema propuesto, será de gran aporte a la entidad, pues los servicios que brindan serán de calidad lo que influirá en la imagen institucional frente a la sociedad en la que se desenvuelve. Además, será un referente para el desarrollo de nuevos sistemas de información a nivel local de las escuelas deportivas.

Marco Teórico

2.1. Introducción

El Capítulo describe conceptos, definiciones que son motivo de estudio para el desarrollo de la propuesta presentada.

Empieza con las definiciones propias del ámbito deportivo, con referencia a la especialidad de fútbol. Posteriormente se definirá conceptos básicos sobre la Ingeniería Web, contemplando fases y procesos que ésta implica.

2.2. Administración Deportiva

La Administración Deportiva no es más que la aplicación del proceso administrativo utilizado por la Administración de Empresas en General (Planear, organizar, dirigir, ejecutar y controlar), así como el uso, e implementación de todas aquellas habilidades humanas, técnicas y conceptuales que necesita el administrador para llevar a cabo una gestión efectiva. Adquiere un carácter deportivo cuando esas habilidades de Administración y Gestión, son aplicadas en aquellas entidades e instituciones deportivas que lo requieren para la consecución pertinente de sus objetivos, igualmente para cuando se hace necesario desarrollar programas, proyectos, programas y eventos con fines deportivos, de recreación, de actividad física, ocio y de aprovechamiento del tiempo libre.

2.2.1. Funciones del Administrador Deportivo

Planear. Determinar por anticipado los objetivos, jerarquizarlos y detallar planes para alcanzarlos, la planeación puede ser:

- **Estratégica:** es decir, proyectada a largo plazo y definida por la alta gerencia.
- **Táctica:** es decir proyectada a mediano plazo y efectuada a nivel de Departamentos.

- **Operacional:** efectuada para cada tarea, constituida por niveles, programas, procedimientos.

Organizar: (Como proceso administrativo) Agrupar las actividades adecuada y lógicamente, la autoridad debe ejercerse de manera que no promueva conflictos.

Dirigir: Hacer que las cosas marchen y sucedan, dinamizar la organización activa sobre el Talento Humano.

Controlar: Asegurar que los resultados de aquello que se planeó, organizó y dirigió se ajusten tanto como sea posible a los objetivos, previamente establecidos.

Evaluar: Seguir un proceso que dé cuenta del quehacer cotidiano y retroalimente la Gestión en forma permanente.

2.3. Gestión Deportiva

Por su parte, GESTIONAR es poner en marcha lo planificado durante la administración. Se puede gestionar un proceso de innovación y desarrollo, un plan de marketing operativo, un sistema estandarizado de subida de archivos a la nube para todos los departamentos de la empresa deportiva.

2.4. Seguimiento Técnico

Seguimiento Técnico, es la observación de la evolución y desarrollo de un proceso deportivo, siendo este proceso planificado y complejo que organiza cargas de trabajos progresivos y crecientes destinadas al favorecimiento del desarrollo de las diferentes capacidades técnicas y cualidades físicas, con el objetivo de promover y consolidar el rendimiento deportivo.

2.5. Rendimiento Deportivo

2.5.1. Planificación Deportiva

La *Planificación del Entrenamiento (deportivo)* es un instrumento fundamental en la gestión del rendimiento deportivo y en **especial en fútbol**, ya que las estructuras de la planificación, las formas de organización del entrenamiento y sus contenidos conforman una estrecha relación con la dinámica de rendimiento pretendida.

Es imposible mejorar el rendimiento en la especialidad del **fútbol** y en cualquier otra especialidad si se entrena siempre lo mismo durante todo el tiempo. Si así fuera, el organismo se adaptaría al entrenamiento que se estuviera realizando y el rendimiento quedaría estático, mientras se mantuviese el mismo régimen de entrenamiento.

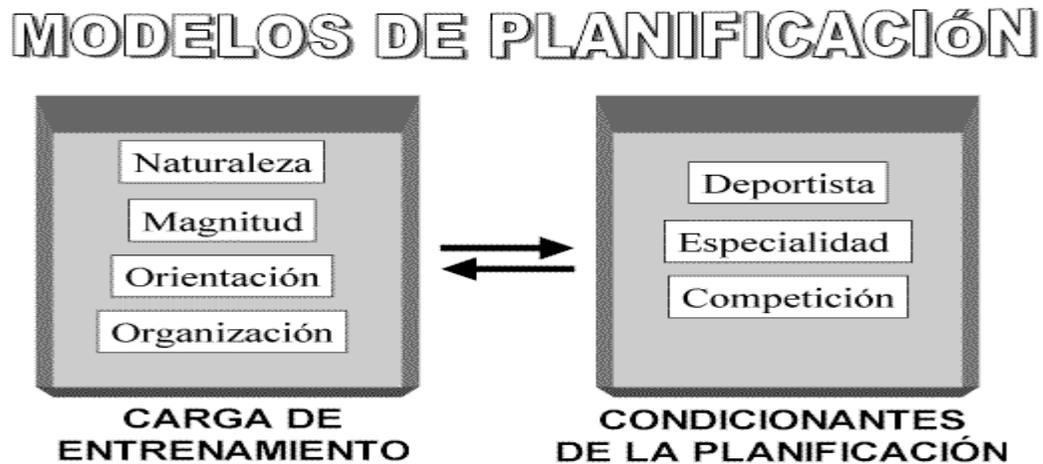
El entrenamiento periodizado es, en esencia, es “*un plan de entrenamiento que cambia las tareas de entrenamiento en un intervalo regular de tiempo*” (Fleck, S. J., & Kraemer, W. J., 1996). Numerosos estudios específicos han mostrado claramente como los programas de entrenamiento periodizado producen mayores ganancias de rendimiento.

Cuando se planifica el entrenamiento con vista a alcanzar el máximo rendimiento posible en una especialidad en un momento determinado, se intenta lograr los niveles de adaptación más elevados posibles de las capacidades relevantes de la especialidad. El procedimiento que se sigue para entrenar estas capacidades está afectado por la forma en que se intenta lograr la adaptación mediante la aplicación de las cargas de entrenamiento en función de su *naturaleza, magnitud, orientación y organización*. Estas cuatro vertientes fundamentales están condicionadas, a su vez, por las *características individuales del deportista, la especialidad deportiva* hacia la que se enfoque el máximo rendimiento y el sistema de competiciones que esta lleve implícita para el logro de los mejores resultados.

Según sea la "forma" en que se aplique la carga de entrenamiento y según se relacionen con los condicionantes de entrenamiento para lograr la mejora del rendimiento deportivo, la planificación del entrenamiento podrá llevarse a cabo mediante diversas opciones o modelos de planificación.

Atendiendo a lo señalado anteriormente, conviene como aspecto previo, describir sucintamente las componentes de la carga de entrenamiento y lo que hemos denominado condicionantes del entrenamiento.

Figura 2- 1:Modelo de Planificación



Fuente: (Fleck, S. J., & Kraemer, W. J., 1996)

2.5.2. Componentes de la Carga de Entrenamiento

La Naturaleza de la Carga (Esfuerzo Físico). Implica lo que se va a trabajar. Viene determinada por el nivel de especificidad y el potencial de entrenamiento (Verjoshanskij, 1990).

Nivel de especificidad. Indica la mayor o menor similitud del ejercicio con la manifestación propia del movimiento durante la competición. En función del nivel de

especificidad de la carga, la carga de entrenamiento se ha identificado tradicionalmente como carga general o carga específica.

Potencial de entrenamiento. Es la forma en que la carga estimula la condición del deportista. Se reduce con el incremento de la capacidad de rendimiento, por lo que es necesario variar los ejercicios o su intensidad para continuar aumentando el rendimiento.

La Magnitud de la Carga. Es el aspecto cuantitativo del estímulo utilizado en el entrenamiento y está determinada por la importancia del volumen, intensidad y duración del entrenamiento exigidos a los deportistas (Verjoshanskij, 1990).

El Volumen. El volumen de la carga es la medida cuantitativa de las cargas de entrenamiento de diferente orientación funcional que se desarrollan en una unidad o ciclo de entrenamiento. Puede ser:

- **Global**, cuando se cuantifica el volumen de todas las cargas de diferente orientación funcional
- **Parcial**, si el volumen de la carga se refiere a un determinado tipo de entrenamiento con una orientación funcional determinada.

La Intensidad. La Intensidad de la carga se entiende como el aspecto cualitativo de la carga ejecutado en un periodo determinado de tiempo. De éste modo, a más trabajo realizado por unidad de tiempo, mayor será la intensidad. La *duración de la carga* es el periodo de influencia de un solo estímulo o un periodo más largo en el que se trabaja con cargas de una misma orientación. Existe evidencia científica de que "existe límite a partir del cual la carga no ejerce más una acción de desarrollo y solo significa una pérdida inútil de tiempo y energía" (Verjoshanskij, 1990).

La Orientación de la Carga. Está definida por la cualidad o capacidad que es potenciada (en el plano físico, técnico, táctico) y por la fuente energética solicitada

predominantemente (procesos aeróbicos, o anaeróbicos). Esta orientación puede ser clasificada en: selectiva y compleja.

La Organización de la Carga. Consiste en la sistematización de la carga en un periodo de tiempo dado con el fin de conseguir un efecto acumulado positivo de las cargas de diferente orientación. Se deben atender dos aspectos: la distribución de las cargas en el tiempo y la interconexión de las cargas (Verjoshanskij, 1990).

Distribución de las cargas. *En el tiempo* es la forma en que se colocan las diferentes cargas en una **sesión, día, microciclo, mesociclo o macrociclo**. Si la carga es distribuida con una única orientación funcional, los medios de entrenamiento se pueden repartir uniformemente en el ciclo (*cargas regulares o diluidas*) o concentrarse en fases definidas del ciclo anual (*cargas concentradas*).

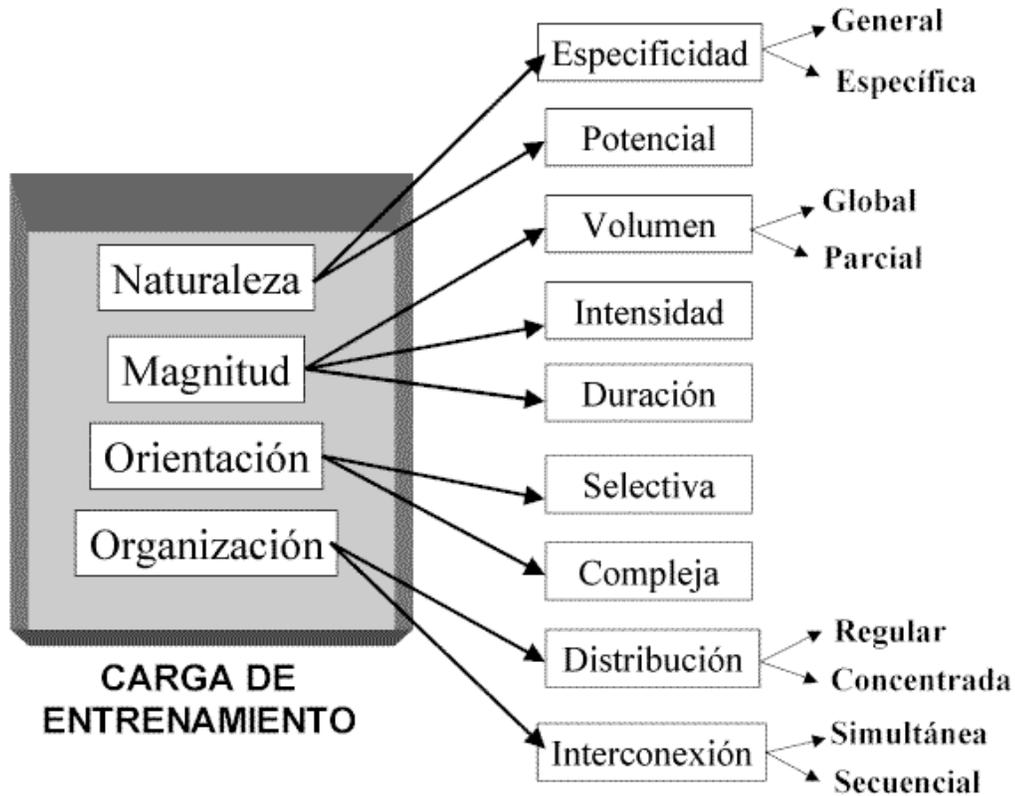
2.5.3. Condicionantes del Entrenamiento

El desarrollo de las diversas capacidades que intervienen en el rendimiento deportivo está condicionado por condiciones que a continuación se detalla.

Por tanto, cada deportista necesitaría su propia planificación. El entrenador que planifica es como un "doctor del entrenamiento", es decir, puede diagnosticar los problemas del paciente (deportista) y sabe (y puede evaluar) las distintas terapias (modelos de planificación) disponibles.

La Especialidad Deportiva. Las capacidades de rendimiento y su importancia relativa en cada especialidad deportiva pueden variar de manera significativa. Una especialidad deportiva en la que domine la fuerza explosiva como capacidad relevante de rendimiento exige una distribución y atención en los contenidos de entrenamiento durante la preparación básica, específica y competitiva distintas a si la especialidad deportiva fuera de resistencia de larga duración (Navarro, F. (1996). Hacia una conceptualización, 1996).

Figura 2- 2:Carga de Entrenamiento



Fuente: (Verjoshanskij, 1990)

Las Condiciones de Competición. En el deporte actual y en especial de fútbol, se ha experimentado un sensible aumento del número de competiciones importantes, introducidas de una forma regular a lo largo de toda la temporada deportiva, con la consecuente necesidad de los deportistas de disponer de un permanente estado de "buen nivel deportivo".

La aplicación de los distintos componentes de la carga de entrenamiento, frente a los diversos condicionantes expuestos, plantean la necesidad de establecer

formulaciones teóricas que se puedan ajustar a la práctica de las distintas alternativas de planificación del entrenamiento.

2.6. Componentes de la Planificación deportiva

Constituye una forma de ordenar los conocimientos e ideas con el objetivo de organizar y desarrollar las sesiones de entrenamiento durante la temporada y que éstas reúnan todos los aspectos propios del juego: tácticos- físicos- psicológicos, teniendo en cuenta el calendario de competición (BOMPA, T.O. (2000): Periodización del entrenamiento, 2000).

Los objetivos de la planificación son los siguientes:

- Evitar la improvisación, que de la misma nace el fracaso.
- Conseguir una continua progresión y mantenimiento de los deportistas.
- Conseguir el máximo estado de forma en la época de la temporada que más nos interese.
- Mantener un estado de forma óptimo durante la competición.

El proceso de entrenamiento se estructura en periodos temporales bien definidos (BOMPA, T.O. (2000): Periodización del entrenamiento, 2000).

2.6.1. Estructura de Proceso de Entrenamiento

El proceso de entrenamiento se estructura en periodos temporales bien definidos (Bompa, 2000):

Macrociclos (ciclos anuales: preparatorio, precompetitivo, competitivo y transición). Constituye un conjunto de varios periodos, describe ciclos de entrenamiento con una duración de 5-6 semanas a 3-4 años. Tiene varias fases:

Adquisición. Periodo Preparatorio (General y específico).

Estabilización. Periodo Competitivo.

Pérdida Temporal. Periodo Transición.

Mesociclos (ciclos de 2 a 8 semanas). Constituye un conjunto de varios microciclos, describe ciclos de entrenamiento de entre dos a ocho semanas. Existe diferentes tipos de mesociclos empleados para la planificación del entrenamiento.

Microciclos (ciclos de aproximadamente una semana). Es la estructura que adapta el entrenamiento en un intervalo de tiempo reducido (7-10 días).

Se caracterizan por:

Acumular y reunir un conjunto de sesiones de entrenamiento cuyos componentes (volumen e intensidad) fluctúan durante dicho periodo.

El grado de carga varía en función de las características de cada sujeto.

Mantienen una estructura de carga creciente y variable.

Tiene como fundamento el principio de la relación óptima entre la carga y la recuperación.

Tipos de Microciclos

Iniciación o introductorio: para iniciar un periodo.

Carga: Se utiliza para plantear cargas importantes y superar los niveles de rendimiento.

Choque: Se plantean para superar grandes cargas de entrenamiento y buscar efectos de choque en el rendimiento.

Acumulación: Para continuar o acumular determinadas cargas del entrenamiento.

Competición: Para preparar una competición importante y obtener un pico en el rendimiento.

Recuperación o regeneración: para restablecer o recuperarse de periodos de carga.

Unidades de entrenamiento (cada sesión en particular). La unidad elemental dentro de una estructura de entrenamiento

Tabla 2- 1 Tipos de Mesociclos

Tipos	Objetivos principales
Introdutorio	Revisión y restablecimiento de la formación deportiva general.
Básico	Mejora del nivel funcional de los factores individuales del rendimiento que se desarrollaron en un nivel superior, en un nivel nuevo, mas alta y compleja capacidad de rendimiento; control de nivel de rendimiento.
Preparatorios y de control	Transformación de los factores individuales del rendimiento que se desarrollaron en un nivel superior, en un nivel nuevo, más alta y compleja capacidad de rendimiento; control de nivel de rendimiento.
Complementario	Eliminación de debilidades conocidas, corrección de errores, estabilización de destrezas y capacidades recién adquiridas.
Competitivo	Se pone especial énfasis sobre la estructuración del rendimiento; trabajo sobre tácticas flexibles y estandarizadas bajo condiciones competitivas.

Intermedio- (recuperación- preparatorio)	a Estructuración ulterior de las bases que determinan el rendimiento después de una serie de competiciones; eliminación de los puntos débiles.
Intermedio- (recuperación- mantenimiento)	b Restauración y estabilización del rendimiento durante un periodo competitivo relativamente largo.
Precompetitivo	Expresión óptima del rendimiento y de todos los factores del rendimiento y su estabilización; preparación para el pico competitivo.
Fuente: (HARRE, 1987)	

2.7. Sistema

Un sistema es un conjunto de elementos relacionados entre sí que funciona como un todo. Si bien cada elemento puede funcionar de manera independiente siempre formará parte de una estructura mayor.

Sistema Deportivo, es el conjunto de todos los practicantes y de todos los servicios deportivos en un territorio determinado. En consecuencia, el fenómeno deportivo puede ser considerado como el "producto" del sistema deportivo. Esto permite su conocimiento y desarrollo de un modo científico actuando sobre todos sus elementos a través de sus relaciones internas y, externamente, mediante las relaciones con los otros sistemas que forman la estructura social."

2.8. Sistema de Gestión

Un sistema de gestión es una herramienta que te permite optimizar recursos, reducir costes y mejorar la productividad en tu empresa. Este instrumento tiene la finalidad reportar datos en tiempo real que permitan tomar decisiones para corregir fallos y prevenir la aparición de gastos innecesarios.

Los sistemas de gestión están basados en normas internacionales que permiten controlar distintas facetas en una empresa o entidad, como la calidad de su producto o servicio. Un sistema de gestión está recomendado a cualquier tipo de organización o actividad orientada a la producción de bienes o servicios, que necesiten mejorar las actividades de la empresa o entidad.

2.9. Ingeniería de software

Es una tecnología con varias capas, cualquier enfoque de ingeniería (incluso la de software) debe basarse en un compromiso organizacional con la calidad. La administración total de la calidad, Six Sigma y otras filosofías similares alimentan la cultura de mejora continua, y es esta cultura la que lleva en última instancia al desarrollo de enfoques cada vez más eficaces de la ingeniería de software. El fundamento en el que se apoya la ingeniería de software es el compromiso con la calidad.

2.9.1. El proceso de software

Forma la base para el control de la administración de proyectos de software, y establece el contexto en el que se aplican métodos técnicos, se generan productos del trabajo (modelos, documentos, datos, reportes, formatos, etc.), se establecen puntos de referencia, se asegura la calidad y se administra el cambio de manera apropiada.

Los métodos de la ingeniería de software. Proporcionan la experiencia técnica para elaborar software. Incluyen un conjunto amplio de tareas, como comunicación, análisis de los requerimientos, modelación del diseño, construcción del programa, pruebas y apoyo. Los métodos de la ingeniería de software se basan en un conjunto

de principios fundamentales que gobiernan cada área de la tecnología e incluyen actividades de modelación y otras técnicas descriptivas.

Las herramientas de la ingeniería de software. Proporcionan un apoyo automatizado o semi automatizado para el proceso y los métodos. Cuando se integran las herramientas de modo que la información creada por una pueda ser utilizada por otra, queda establecido un sistema llamado ingeniería de software asistido por computadora que apoya el desarrollo de software.

Figura 2- 3 Capas de la Ingeniería de Software



Fuente: (Pressman, 2010)

2.9.2. Aplicaciones Web

Una Aplicación Web (web app) es una versión de la página web optimizada y adaptable a cualquier dispositivo móvil. Dicho de otra manera, es una página que se puede abrir desde el navegador de cualquier terminal independientemente del sistema operativo que utilice. Esta optimización es posible gracias a HTML5 y CSS3.

2.10. Metodología UWE

El UWE, consiste en la construcción de modelos y análisis y diseño. Dicha construcción se realiza dentro de un marco de un proceso de diseño e iterativo e incremental. Las actividades del modelado abarcan: el análisis de requerimientos, diseño conceptual, diseño de usuario diseño de la navegación, diseño de la presentación y diseño de la adaptación.

Los principales modelos que produce el método UWE son los siguientes:

- Un modelo de requerimientos, que captura los requerimientos del sistema.
- Un modelo conceptual, para el contenido (modelo de contenido).
- Un modelo de usuario.
- Un modelo de navegación, que comprende la estructura de la navegación.
- Un modelo de presentación, que abarca modelos estáticos y dinámicos (modelo de estructura de la presentación, modelo del flujo de la presentación, modelo de interface abstracta del usuario y modelo de ciclo de vida del objeto.)
- Un modelo de adaptación.

2.10.1. Modelo de Requerimientos

El primer paso para el desarrollo del sistema Web, especificado con UWE, es realizar la identificación de requerimientos y plasmarlos en un modelo de requerimientos.

Los modelos pueden ser documentados en diferentes niveles de detalle, en este caso UWE, especifica dos niveles de granularidad, la primera es describir

detalladamente las funcionalidades del sistema, especificadas en diagramas de casos de uso UML. Y como segundo paso se debe elaborar una descripción de los casos de uso más detallada usando diagramas de actividad especificando responsabilidades y acciones de los actores involucrados.

2.10.2. Modelo de Contenido (conceptual) y modelo de usuario

El diseño conceptual se basa en el modelo de análisis e incluye objetos involucrados en las típicas que los usuarios realizan con la aplicación.

El propósito del modelo de contenido es proporcionar una especificación visual de la información relevante para el dominio del sistema web, que comprende principalmente contenido de la aplicación web.

2.10.3. Modelo de Navegación

El modelo de estructura de navegación define la estructura de nodos y links de una WebApp mostrando cómo se puede realizar la navegación utilizando elementos de acceso tales como índice, visitas guiadas consultas y menús

Los elementos del modelado son:

Clases de navegación, que se denotan con (0), representan nodos navegables de la estructura de hipertexto.

Links de navegación, que muestran el vínculo directo entre clases de navegación.

Caminos de navegación alternativos, los cuales son visualizados con el estereotipo <<menu>>().

Primitivas de acceso. Las cuales se utilizan ya sea para llegar a múltiples instancias de una clase de navegación (<<index>>) o para seleccionar ítems (<<query>>).

Clases de procesos (), las cuales modelan los puntos de entrada y de salida de procesos de negocio. Cada clase de proceso está asociada a un caso de uso de proceso.

Links de procesos, que representan el vínculo entre las clases de proceso y de navegación.

El modelo de estructura de navegación se representa mediante diagrama de clases UML estereotipado con clases de navegación y procesos, menús y primitivas de acceso y así también los links de navegación y de proceso.

2.10.4. Modelo de Presentación

El modelo de presentación proporciona una vista abstracta de la interfaz de usuario (UI) de la aplicación web, se basa en el modelo de navegación y describe que elementos (por ejemplo, texto, links, formularios) se utilizaran para presentar los nodos de navegación.

2.11. Ciclo de Vida del Software

El ciclo de vida del software, son fases por las que pasa el sistema que se está desarrollando desde que nace la idea inicial hasta que el software es retirado o reemplazado.

Entre las funciones que debe tener un ciclo de vida se pueden destacar:

Determinar el orden de las fases del proceso de software.

Establecer los criterios de transición para pasar de una fase a la siguiente.

Definir las entradas y salidas de cada fase.

Describir los estados por lo que pasa cada producto.

Describir las actividades a realizar para transformar el producto.

Definir un esquema que sirve como base para planificar, organizar, coordinar y desarrollar.

2.11.1. Modelo Iterativo e Incremental

En un desarrollo iterativo e incremental el proyecto se planifica en diversos bloques temporales, entendiéndose como mini proyectos. En todas las iteraciones se repite un proceso de trabajo similar para proporcionar un resultado completo sobre el producto final, de manera que el cliente pueda obtener los beneficios del proyecto de forma incremental.

En cada iteración el equipo evoluciona el producto (hace una entrega incremental) a partir de los resultados completados en las iteraciones anteriores añadiendo nuevos objetivos/requisitos o mejorado los que ya fueron completados. Un aspecto fundamental para guiar el desarrollo iterativo incremental es la priorización de los objetivos/requisitos en función del valor que aportan al cliente.

2.12. Orientada a Objetos

Aunque UML puede emplearse en cualquier paradigma, como la programación estructurada o la lógica, está especialmente cerca del paradigma de la orientación a objetos. Por tanto, es precisa una familiarización con algunos detalles de este paradigma antes de continuar con UML.

2.12.1. Objeto

De manera intuitiva, la tendencia general es asociar el término objeto con todo aquello a lo que se puede atribuir la propiedad física de masa, como una tostadora de pan, aunque es posible encontrar objetos de índole no tangible, como por ejemplo una dirección postal. En el ámbito de la informática, un objeto define una representación abstracta de las entidades del mundo, tangibles o no, con la intención de emularlas. Existe pues, una relación directa entre los objetos del mundo y los objetos informáticos, de modo que puede emplearse el término objeto de manera indistinta.

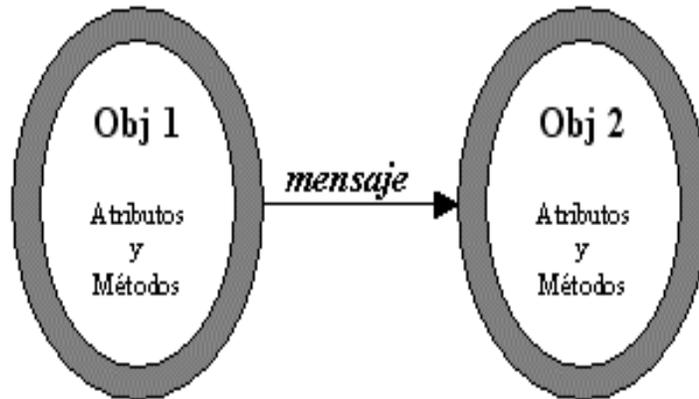
Los objetos tienen dos características, que son su estado y su comportamiento. El estado es una situación en la que se encuentra el objeto, tal que cumple con alguna condición o condiciones particulares, realiza alguna actividad o espera que suceda un acontecimiento. Una tostadora puede estar encendida y cargada de pan y, en cuanto a su comportamiento, lo normal en este estado es tostar pan.

En el envío de mensajes es la forma en que se invocan los métodos de un objeto y que la invocación de métodos es el mecanismo a través del cual un objeto puede cambiar su estado o el de otro objeto. Los atributos y los métodos de un objeto pueden tener un menor o mayor grado de visibilidad, desde “privado” hasta “público”, lo que hace que aparezca un concepto nuevo, la encapsulación. La encapsulación oculta los detalles del funcionamiento interno del objeto, exponiendo sólo aquello que pueda ser de interés.

2.12.2. Clase

Los objetos no son entidades que existan de modo único. Hay muchos tipos de tostadoras e, igualmente, muchas tostadoras del mismo tipo. Se puede entender fácilmente el concepto de clase si nos permitimos emplear el término *tipo* como equivalente. Así, todos los objetos que son del mismo *tipo*, comparten el mismo juego de atributos y métodos (aunque cada objeto pueda tener un valor distinto asociado a cada atributo) y por tanto pertenecen a una misma clase. Las clases son como patrones que definen qué atributos y qué métodos son comunes a todos los objetos de un mismo tipo.

Figura 2- 4 Objetos Comunicándose



Fuente: (Schmuller, 1997)

Cada objeto tiene sus atributos y su comportamiento, creados empleando una clase a modo de patrón. Una vez creado el objeto, pasa a ser una instancia particular de la clase a la que pertenece y sus atributos tienen unos valores concretos, que podrán variar de un objeto a otro (dos objetos distintos pertenecientes a la misma clase, pueden tener exactamente los mismos valores en todos sus atributos). A estos atributos, que pueden variar de un objeto a otro, se les conoce también como *variables de instancia*.

Hay atributos que, sin embargo, no varían de un objeto a otro, es decir todas las instancias de la clase a la que pertenecen, tienen el mismo valor para ese atributo. Todas las tostadoras del mismo tipo consumen los mismos Watios y sus resistencias son de los mismos Ohmios. A estos atributos se les conoce como *variables de clase* y son compartidos por todas y cada una de las instancias de la clase. De manera análoga al caso de los atributos, encontramos *métodos de instancia* y *métodos de clase*.

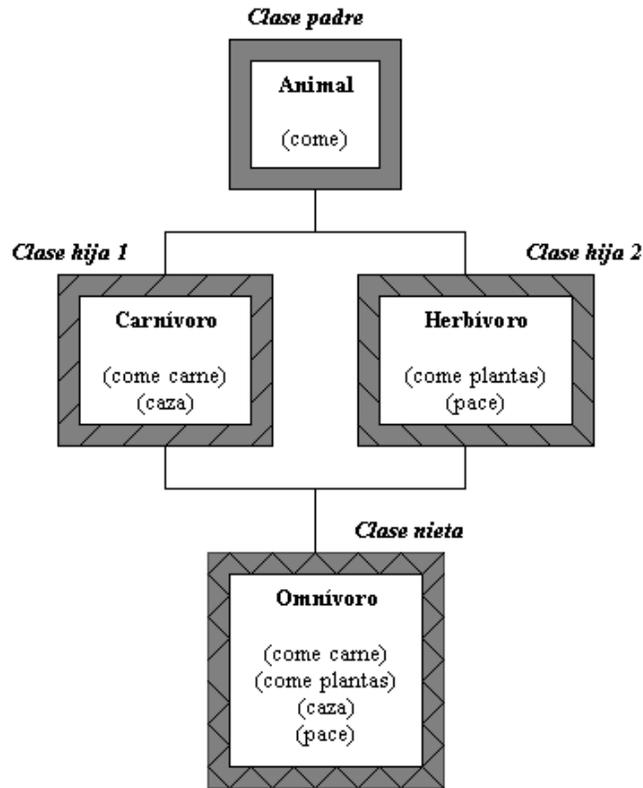
2.12.3 Herencia

Los objetos se definen en función de clases, es decir, tomando una clase como patrón. Se puede saber mucho acerca de un objeto sabiendo la clase a la que pertenece. Por ejemplo, con decir que la “Russell Hobbs 10243 Kudos” es un tipo de tostadora, inmediatamente se sabe que se trata de una máquina para tostar pan, probablemente eléctrica y con por lo menos una ranura en la que insertar una rebanada de pan y un botón para activar su funcionamiento.

Las clases llegan un paso más lejos, permitiendo su definición en función de otras clases, de modo que es posible establecer una jerarquía de especialización. Una clase que se define en función de otra, hereda todos los atributos y métodos de aquella y permite el añadido de nuevos o sobre escritura de los heredados. La clase patrón se conoce con el nombre de *superclase* o *clase padre*, mientras que la que hereda se conoce como *clase hija*. La herencia no está limitada simplemente a padre-hija(s), la jerarquía puede ser todo lo profunda que sea necesario, hablando en términos de nietas, bisnietas, etc. De la misma manera, una clase puede heredar de varias clases a la vez.

En la siguiente figura se puede ver una jerarquía de especialización de dos niveles. La clase “Animal” es la raíz, la clase padre en la jerarquía. Especifica que los animales *comen*, como característica más significativa de éstos. En el primer nivel de especialización encontramos las clases “Carnívoro” y “Herbívoro”, ambas son sendos tipos de animal y por tanto *comen*, sólo que en el caso de los carnívoros se ha especializado el tipo de comida que comen para indicar que se trata de *carne*. Aparece una nueva característica de este tipo de animal, que es el hecho de que los carnívoros *cazan*. En el caso de los herbívoros, encontramos que *comen plantas* y *pacen*. En el segundo nivel de especialización, encontramos un animal que es a la vez “Herbívoro” y “Carnívoro” y, como cabe esperar, este nuevo tipo de animal puede hacer todo lo que pueden hacer sus ancestros, comer carne, comer plantas, cazar y pacer, no encontrando ninguna característica adicional en los “Omnívoros”.

Figura 2- 5 Ejemplo Herencia



Fuente: (Schmuller, 1997)

2.12.4. Interfaz

Una interfaz es un mecanismo que emplean dos objetos para interactuar. En nuestro ejemplo de la tostadora, el humano emplea el botón de tostar a modo de interfaz para pasar el mensaje “tuesta el pan que tienes en la bandeja”.

Las interfaces definen un conjunto de métodos para establecer el protocolo en base al cual interactúan dos objetos. En este sentido, existe una analogía entre

interfaces y protocolos. Para que el humano pueda tostar, debe seguir el protocolo establecido por la interfaz botón de tostar, consistente en pulsar dicho botón.

Las interfaces capturan las similitudes entre clases no relacionadas, sin necesidad de forzar una interrelación y son a su vez clases.

2.13. Diagramas UML

UML es un lenguaje gráfico que sirve para modelar, diseñar, estructurar, visualizar, especificar, construir y documentar software. UML proporciona un vocabulario común para toda la cadena de producción, desde quien recaba los requisitos de los usuarios, hasta el último programador responsable del mantenimiento. Es un lenguaje estándar para crear los planos de un sistema de forma completa y no ambigua. Fue creado por el Object Management Group, OMG, un consorcio internacional sin ánimo de lucro, que asienta estándares en el área de computación distribuida orientada a objetos, y actualmente revisa y actualiza periódicamente las especificaciones del lenguaje, para adaptarlo a las necesidades que surgen. El prestigio de este consorcio es un aval más para UML, considerando que cuenta con socios tan conocidos como la NASA, la Agencia Europea del Espacio ESA, el Instituto Europeo de Bioinformática EBI, Boeing, Borland, Motorola y el W3C, por mencionar algunos.

2.13.1. Bloques básicos de construcción de UML

Los bloques básicos de construcción de UML son tres, los elementos, las relaciones y los diagramas.

Los elementos. Son abstracciones que actúan como unidades básicas de construcción. Hay cuatro tipos, los *estructurales*, los de *comportamiento*, los de *agrupación* y los de *notación*. En cuanto a los elementos estructurales son las partes estáticas de los modelos y representan aspectos conceptuales o materiales. Los elementos de comportamiento son las partes dinámicas de los modelos y representan comportamientos en el tiempo y en el espacio. Los elementos de agrupación son las

partes organizativas de UML, establecen las divisiones en que se puede fraccionar un modelo. Sólo hay un elemento de agrupación, el paquete, que se emplea para organizar otros elementos en grupos. Los elementos de notación son las partes explicativas de UML, comentarios que pueden describir textualmente cualquier aspecto de un modelo. Sólo hay un elemento de notación principal, la nota.

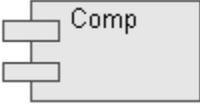
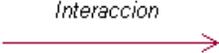
Las relaciones son abstracciones que actúan como unión entre los distintos *elementos*. Hay cuatro tipos, la *dependencia*, la *asociación*, la *generalización* y la *realización*.

Los diagramas son la disposición de un conjunto de elementos, que representan el sistema modelado desde diferentes perspectivas. UML tiene nueve diagramas fundamentales, agrupados en dos grandes grupos, uno para modelar la estructura estática del sistema y otro para modelar el comportamiento dinámico. Los **diagramas estáticos son:** el de *clases*, de *objetos*, de *componentes* y de *despliegue*. Los **diagramas de comportamiento son:** el de *Casos de Uso*, de *secuencia*, de *colaboración*, de *estados* y de *actividades*.

Tabla 2- 2 Elementos UML

	Clase	<pre> classDiagram class Tostadora { noBandejas color tostar() } </pre>	Describe un conjunto de objetos que comparten los mismos atributos, métodos, relaciones y semántica. Las clases implementan una o más interfaces.
		<pre> classDiagram class Tostadora { noBandejas color tostar() comprobarTemp() } </pre>	Se trata de una clase, en la que existen procesos o hilos de ejecución concurrentes con otros elementos. Las líneas

E L E M E N T O S E S T	Clase activa		del contorno son más gruesas que en la clase “normal”
	Interfaz	 Interfaz	Agrupación de métodos u operaciones que especifican un servicio de una clase o componente, describiendo su comportamiento, completo o parcial, externamente visible. UML permite emplear un círculo para representar las interfaces, aunque lo más normal es emplear la clase con el nombre en cursiva.
	Colaboración	 Colaboracion	Define una interacción entre elementos que cooperan para proporcionar un comportamiento mayor que la suma de los comportamientos de sus elementos.
	Caso de uso	 Caso de Uso	Describe un conjunto de secuencias de acciones que un sistema ejecuta, para producir un resultado observable de interés. Se emplea para estructurar los aspectos de comportamiento de un modelo.

<p>R U C T U R A L E S</p>	<p>Componente</p>		<p>Parte física y por tanto reemplazable de un modelo, que agrupa un conjunto de interfaces, archivos de código fuente, clases, colaboraciones y proporciona la implementación de dichos elementos.</p>
<p>Elementos de comportamiento</p>	<p>Interacción</p>		<p>Comprende un conjunto de mensajes que se intercambian entre un conjunto de objetos, para cumplir un objetivo específico.</p>
	<p>Máquinas de estados</p>		<p>Especifica la secuencia de estados por los que pasa un objeto o una interacción, en respuesta a eventos.</p>

Elementos de agrupación	Paquete		Se emplea para organizar otros elementos en grupos.
Elementos de notación	Nota		Partes explicativa de UML, que puede describir textualmente cualquier aspecto del modelo

Fuente: (Schmuller, 1997)

2.13.2. Diagrama de Clases y Diagrama de Objetos

Los diagramas de clases muestran un resumen del sistema en términos de sus clases y las relaciones entre ellas. Son diagramas estáticos que muestran **qué** es lo que interactúa, pero no cómo interactúa o qué pasa cuando ocurre la interacción.

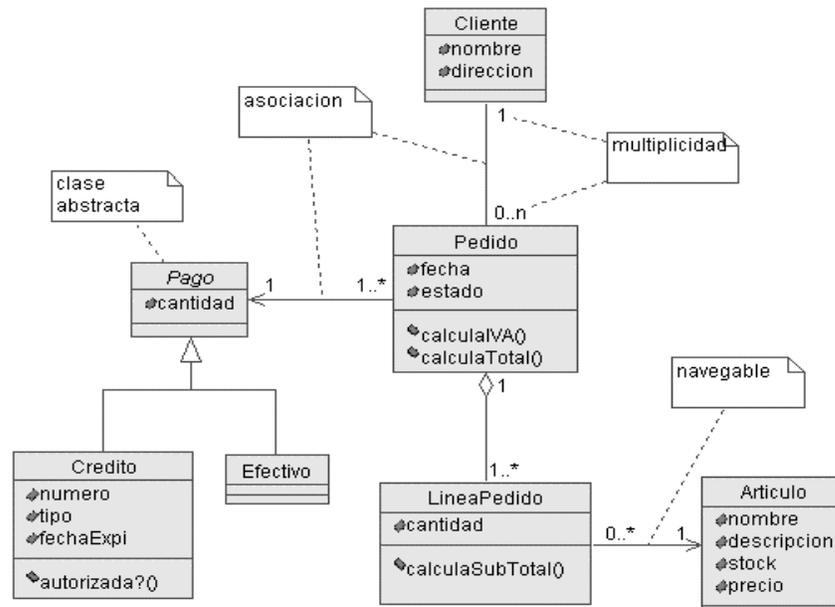
El siguiente diagrama modela los pedidos de un cliente a una tienda de venta por catálogo. La clase principal es “Pedido”, asociada a un cliente, una forma de pago y un conjunto de artículos.

La clase “Pago” es abstracta, en UML los nombres de clases abstractas se representan en *Itálica*. Las clases abstractas actúan a modo de **interfaz**, proporcionando únicamente un listado de métodos a ser “realizados” por las clases que las implementan o realizan. “Pago” es una superclase especializada, y a la vez realizada, por sus formas más comunes “Credito” y “Efectivo”. Un “Pedido” tiene una

única forma de pago, expresada por su multiplicidad, 1, mientras que una forma de pago puede estar presente en uno o más pedidos, como sugiere su multiplicidad, 1..*.

En cuanto a las asociaciones, observamos que algunas vienen representadas como una *flecha navegable*, cuya orientación expresa el sentido en que se consultan los datos. Las asociaciones sin flecha son bi-direccionales. Las agregaciones expresan “conjunto de”; la relación entre “Pedido” y “Artículo” es de conjunto. Un pedido es una agregación de una o más líneas de pedido, donde cada una hace alusión a un artículo concreto, así mismo una línea de pedido puede estar presente en varios pedidos y un artículo puede no haber sido solicitado nunca.

Figura 2- 6 Diagrama de Clases



Fuente: (Booch, Rumbaugh, & Jacobson, 1996)

En cuanto a la multiplicidad, la siguiente tabla resume las más comunes. Hay que tener en cuenta que la multiplicidad se expresa “en el lado opuesto” de la relación y es el número de posibles instancias de una clase asociadas con una única instancia de la clase en el otro extremo.

Tabla 2- 3 Multiplicidad de Diagrama de Clases

Multiplicidad	Significado
1	Una única instancia
N / *	N instancias
0..N / 0..*	Entre ninguna y N instancias
1..N / 1..*	Entre una y N instancias
0..1	Ninguna o una instancia
N..M	Entre N y M instancias

Fuente: (Booch, Rumbaugh, & Jacobson, 1996)

Los diagramas de objetos son análogos a los de clases, con la particularidad de que, en lugar de encontrar clases, encontramos instancias de éstas. Son útiles para explicar partes pequeñas del modelo en las que hay relaciones complejas.

Tabla 2- 4 Relaciones UML

Dependencia		Es una relación entre dos elementos, tal que un cambio en uno puede afectar al otro.
Asociación		Es una relación estructural que resume un conjunto de enlaces que son conexiones entre objetos.

Generalización		Es una relación en la que el elemento generalizado puede ser substituido por cualquiera de los elementos hijos, ya que comparten su estructura y comportamiento.
Realización		Es una relación que implica que la parte realizante cumple con una serie de especificaciones propuestas por la clase realizada (interfaces).

Fuente: (Schmuller, 1997).

2.13.3. Diagrama de Casos de Uso

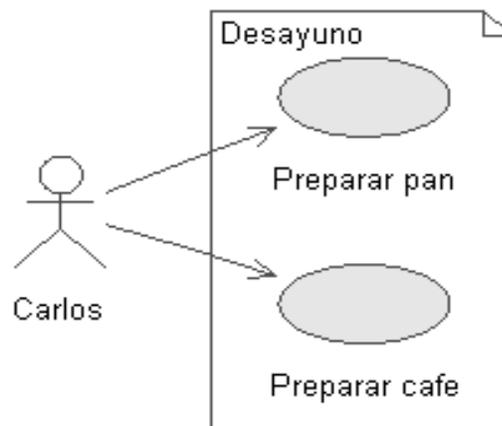
Los diagramas de Casos de Uso describen lo que hace un sistema desde el punto de vista de un observador externo, enfatizando el **qué** más que el cómo. Plantean escenarios, es decir, lo que pasa cuando alguien interactúa con el sistema, proporcionando un resumen para una tarea u objetivo. El siguiente Caso de Uso describe como Carlos va a desayunar (este es su objetivo), para lo que se plantea el escenario de preparar su café y el pan tostado.

En los Casos de Uso, los *Actores* son papeles que determinadas personas u objetos desempeñan. Se representan mediante un “hombre de palitos”, de modo que, en el ejemplo, Carlos es un Actor. Los Casos de Uso se representan por medio de *óvalos* y las líneas que unen Actores con Casos de Uso representan una asociación de comunicación.

Por su puesto, un Caso de Uso puede ser descrito en mayor profundidad. Por ejemplo, si tomamos por separado “Preparar pan” y “Preparar cafe”, podemos bajar un nivel de descripción y llegar a los siguientes Casos de Uso.

El siguiente Caso de Uso es equivalente al primero, “Desayuno”, sólo que en él se ha condensado la máxima cantidad posible de información. En él se muestra un nuevo elemento que hasta ahora no se había mostrado, el “**estereotipo**”, que viene entre sendos símbolos angulados “<<” y “>>” y concreta un paso más allá el tipo de relación existente entre dos Casos de Uso. Encontramos dos estereotipos <<include>> y <<extend>>. El primero indica que el Caso de Uso “Tostar pan” requiere de “Usar tostadora” para poder ser llevado a cabo. Esta es una forma muy adecuada de sacar factor común entre Casos de Uso, o incluso de fraccionar Casos de Uso muy grandes. El segundo indica que el Caso de Uso “Untar pan” es una variación de “Untar”. Observamos también que “Comer pan” y “Beber cafe” son una generalización de “Alimentarse”.

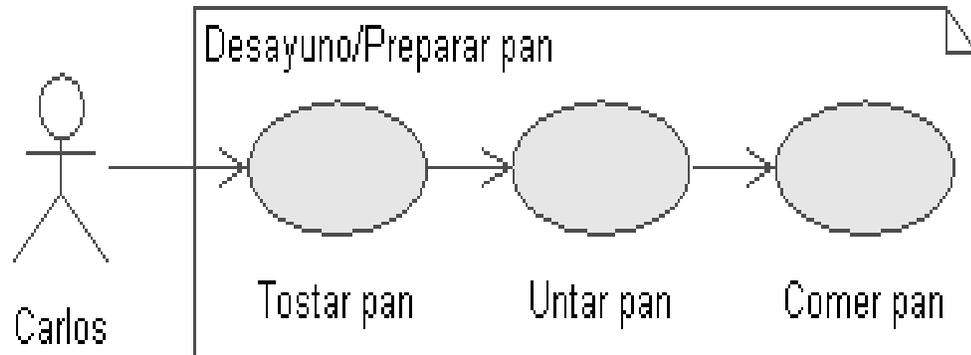
Figura 2- 7 Diagrama de Casos de uso



Fuente: (Booch, Rumbaugh, & Jacobson, 1996)

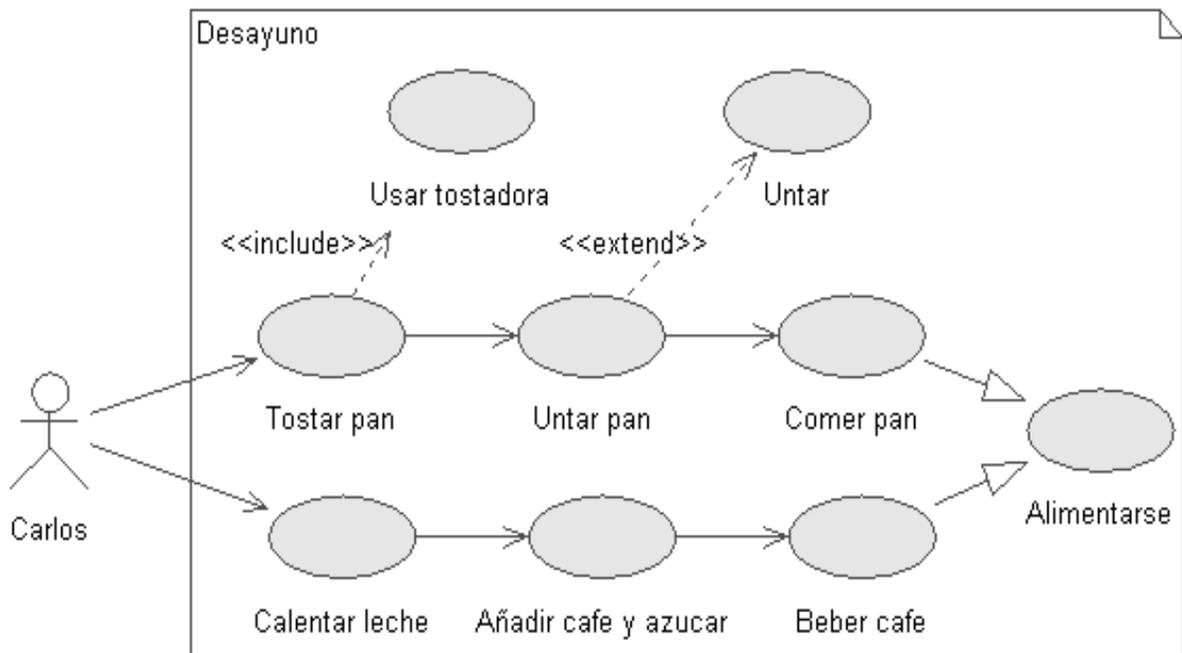
Los Casos de Uso suelen venir delimitados por fronteras o límites, que definen una separación entre lo que es realmente la funcionalidad del sistema y los actores que la usan o colaboran en su desempeño. En las figuras, esta separación viene representada por medio de la caja que encapsula los óvalos.

Figura 2- 8 Caso de uso Nivel 2



Fuente: (Booch, Rumbaugh, & Jacobson, 1996)

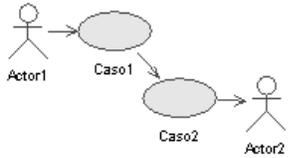
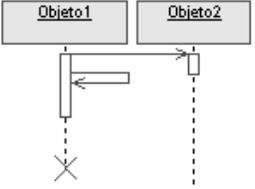
Figura 2- 9 Diagrama de Caso de Uso Extendido



Fuente: (Booch, Rumbaugh, & Jacobson, 1996)

“Carlos va a desayunar. Para ello debe hacer dos actividades distintas, pero relacionadas. La primera consiste en tostar pan, para lo cual necesita emplear una tostadora. Una vez tostado el pan, lo unta de mantequilla y mermelada de fresa (untar pan no es muy distinto de untar otro tipo de alimentos). La segunda consiste en preparar el café, por lo cual necesita calentar leche y añadir café y azucar. Terminadas ambas actividades, Carlos puede proceder a alimentarse, comiendo el pan tostado y bebiendo el café. El orden en que realice las actividades da igual y también da igual si se realizan a la vez.”

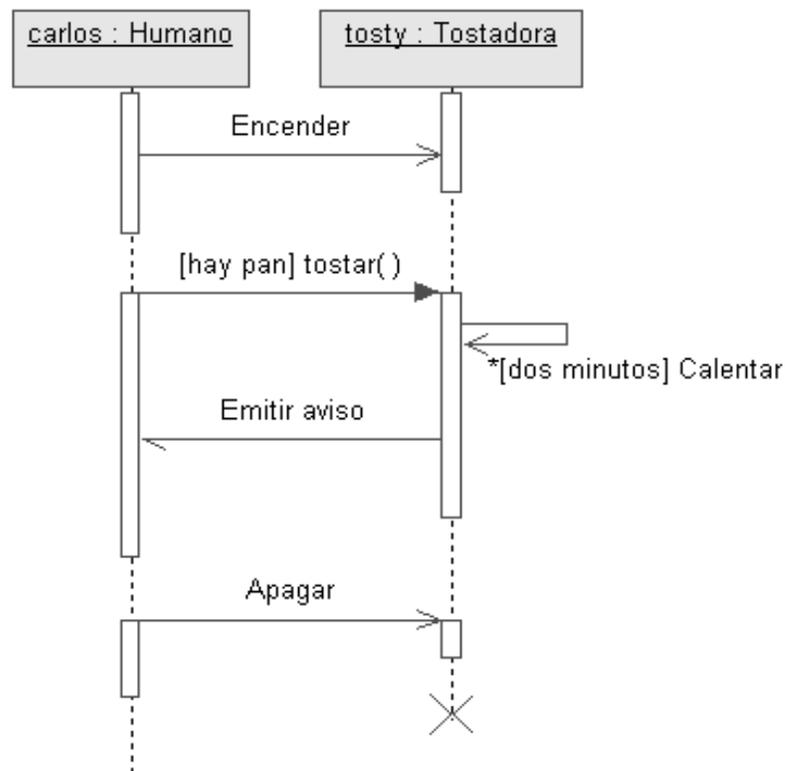
Tabla 2- 5 Diagramas UML

M O D E L A	Casos de Uso	 <pre> graph LR Actor1((Actor1)) --> Caso1([Caso1]) Caso1 --> Caso2([Caso2]) Caso2 --> Actor2((Actor2)) </pre>	Muestra un conjunto de casos de uso, los actores implicados y sus relaciones. Son diagramas fundamentales en el modelado y organización del sistema.
	Secuencia	 <pre> sequenceDiagram participant Objeto1 participant Objeto2 Objeto1->>Objeto2: Objeto2-->>Objeto1: destroy Objeto1 </pre>	Son diagramas de interacción, muestran un conjunto de objetos y sus relaciones, así como los mensajes que se intercambian entre ellos. Cubren la vista dinámica del sistema. El diagrama de secuencia resalta la ordenación temporal de los mensajes, mientras que el de colaboración resalta la organización estructural de los objetos, ambos siendo equivalentes o isomorfos. En el diagrama de colaboración de la figura de la izquierda, se puede ver que los elementos gráficos no son cajas rectangulares, como cabría

2.13.4. Diagrama de Secuencia y Diagrama de Colaboración

Los diagramas de secuencia describen como los objetos del sistema colaboran. Se trata de un diagrama de interacción que detalla como las operaciones se llevan a cabo, qué mensajes son enviados y cuando, organizado todo en torno al tiempo. El tiempo avanza “hacia abajo” en el diagrama. Los objetos involucrados en la operación se listan de izquierda a derecha de acuerdo a su orden de participación dentro de la secuencia de mensajes.

Figura 2- 10 Diagrama de Secuencia



Fuente: (Booch, Rumbaugh, & Jacobson, 1996)

Las líneas verticales o “líneas de la vida” representan el tiempo de vida del objeto. La vida del objeto “carlos” no termina en este diagrama, sin embargo, la del

objeto “tosty” sí y esto viene representado mediante el aspa al final de su línea de la vida.

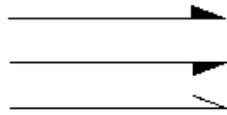
Los rectángulos verticales son barras de activación y representan la duración de la ejecución del mensaje. El mensaje “Encender”, posiblemente implementado mediante la introducción del enchufe en una toma de pared, tiene una duración escasa y similar a la de “Apagar”. No ocurre lo mismo con la llamada al método “tostar()”, que dura desde la pulsación del botón de tostar hasta que el pan es retirado de la bandeja y además interviene la emisión de un aviso cuando el pan está lo suficientemente caliente, a fin de evitar que se quemé.

Como se puede observar, la acción tostar viene condicionada por la presencia de pan en la bandeja de la tostadora. En UML los corchetes “[]” expresan condición y si están precedidos de un asterisco indican interacción mientras se cumpla la condición.

Los mensajes que son intercambiados entre los objetos de un diagrama de secuencia pueden ser *síncronos* o *asíncronos*. Los mensajes asíncronos son aquellos tal que el emisor puede enviar nuevos mensajes mientras el original está siendo procesado. El mensaje asíncrono ocurre en el tiempo de manera independiente a otros mensajes. Los mensajes síncronos son todo lo contrario, el emisor debe esperar a que termine el tiempo de proceso del mensaje antes de que pueda emitir nuevos mensajes. UML emplea los siguientes convenios para representar el tipo de mensaje.

Tabla 2- 6 Tipos de mensajes en diagramas de interacción

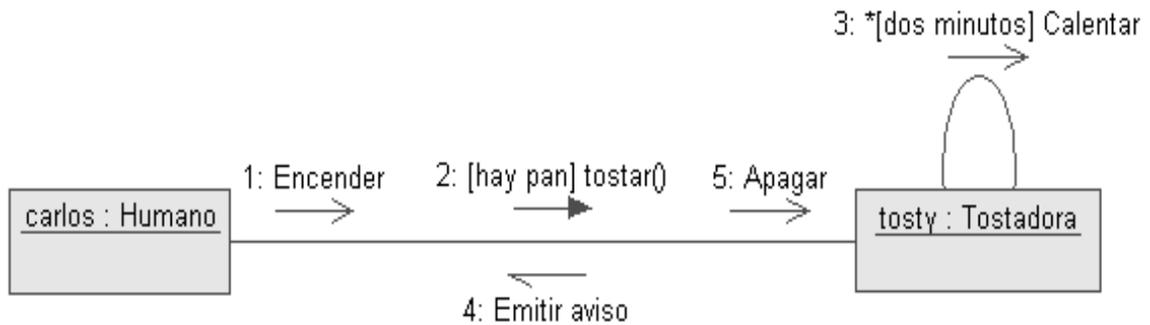
Símbolo	Significado
	Mensaje simple que puede

	ser síncrono o asíncrono.
	Mensaje simple de vuelta (opcional).
	Mensaje síncrono.
	Mensaje asíncrono.

Fuente: (Booch, Rumbaugh, & Jacobson, 1996)

Los diagramas de colaboración son otro tipo de diagramas de interacción, que contiene la misma información que los de secuencia, sólo que se centran en las responsabilidades de cada objeto, en lugar de en el tiempo en que los mensajes son enviados. Cada mensaje de un diagrama de colaboración tiene un número de secuencia. El primer nivel de la secuencia es 1, y los mensajes que son enviados durante la misma llamada a un método se numeran 1.1, 1.2 y así sucesivamente para tantos niveles como sea necesario.

Figura 2- 11 Diagrama de Colaboración



Fuente: (Booch, Rumbaugh, & Jacobson, 1996)

2.13.5. Diagrama de Estados y Diagrama de Actividades

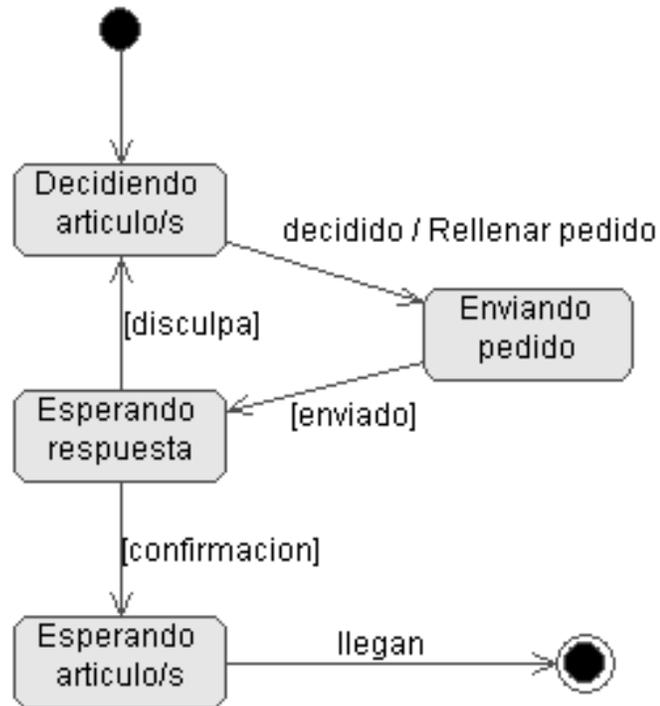
Los diagramas de estados. Muestran los posibles estados en que puede encontrarse un objeto y las transiciones que pueden causar un cambio de estado. El estado de un objeto depende de la actividad que esté llevando a cabo o de alguna condición.

Las transiciones son las líneas que unen los diferentes estados. En ellas se representa la condición que provoca el cambio, seguida de la acción oportuna separada por “/”. En un estado en que el objeto está pendiente de algún tipo de validación que dependa de un proceso en curso, no es necesario evento externo alguno para que se produzca la transición, ya que ésta ocurrirá cuando termine el proceso, en función del resultado de éste. En estos casos es conveniente, por claridad, incluir la condición que de la que depende la transición (entre corchetes).

Los estados iniciales, a partir del que se “entra” en la máquina de estados, y final, que indica que la máquina de estados termina, no tienen otro significado adicional, son elementos ornamentales y se representan mediante un círculo negro y un círculo negro resaltado respectivamente.

Los estados de un diagrama de estados pueden anidarse, de forma que los estados relacionados pueden ser agrupados en un estado compuesto. Esto puede ser necesario cuando una actividad involucra sub-actividades asíncronas o concurrentes.

Figura 2- 12 Máquina de Estados



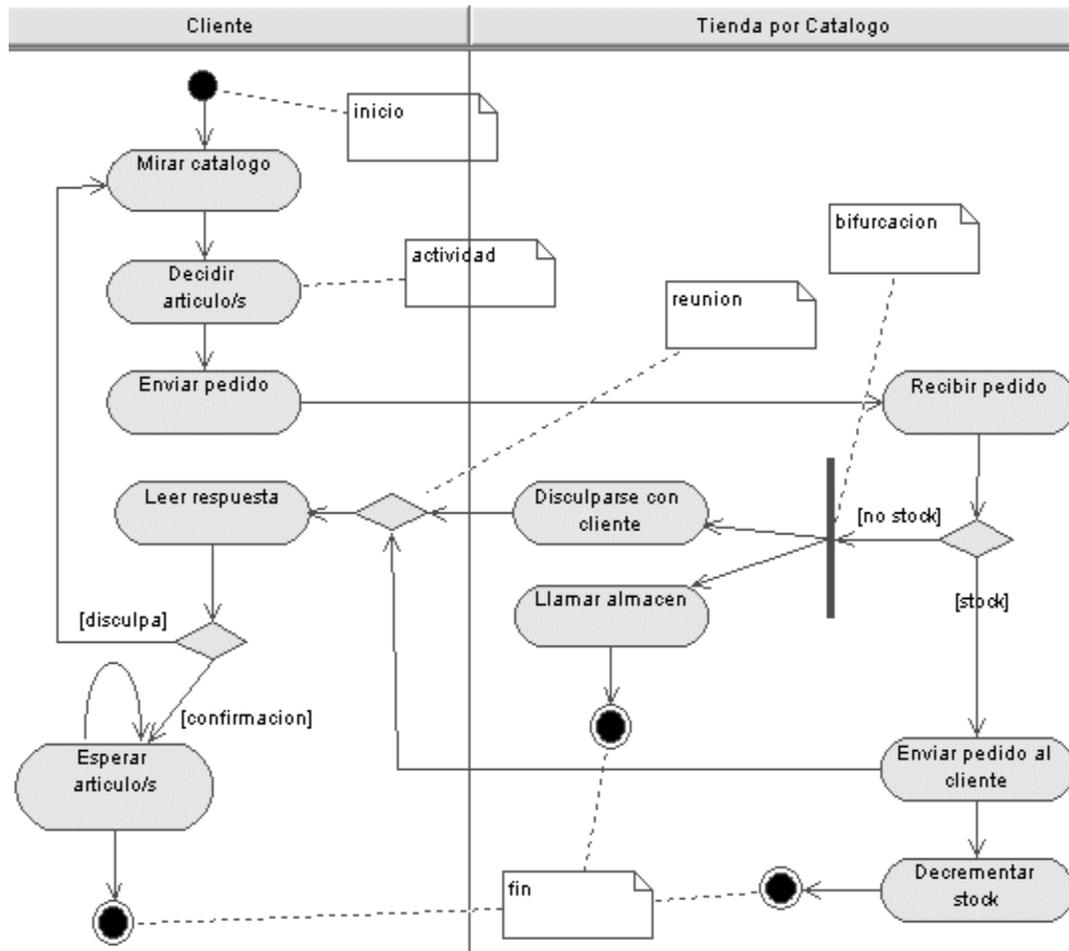
Fuente: (Booch, Rumbaugh, & Jacobson, 1996)

Los diagramas de actividades son básicamente diagramas de flujo adornados, que guardan mucha similitud con los diagramas de estados. Mientras que los diagramas de estados centran su atención en el proceso que está llevando a cabo un objeto, los diagramas de actividades muestran como las actividades fluyen y las dependencias entre ellas.

Los diagramas de actividades pueden dividirse en “calles” que determinan qué objeto es responsable de qué actividad. Las actividades vienen unidas por transiciones, que pueden separarse en ramas en función del resultado de una condición expresada entre

corchetes. Cada rama muestra la condición que debe ser satisfecha para que el flujo opte por ese camino. Igualmente, las transiciones se pueden bifurcarse en dos o más actividades paralelas.

Figura 2- 13 Diagrama de Actividades



Fuente: (Schmuller, 1997)

2.14. Calidad de Software

Desde la década del 90 se comenzó a escuchar la palabra calidad con mayor profundidad en todos los ámbitos de las actividades del ser humano como en la

educación, la industria, los negocios, los productos, hasta ser una condición también del ser humano.

El software no escapa de esta tendencia, el problema radica en cómo definir y tratar la calidad en un elemento de origen abstracto, lógico e intangible, y donde este término se transforma aún más en un concepto complejo y multifacético, que puede describirse desde diversas perspectivas. Para analizar estas distintas visiones determinaremos varios puntos de vista: el del Cliente, el del Usuario, el del Productor, y el del Producto.

Según la IEEE la calidad del software debe:

Ser mensurable y predecible. En cuanto a los factores para lograr dicha calidad la IEEE indica que se deben cumplir los siguientes requisitos:

Ausencia de defectos.

Satisfacción del usuario.

Conformidad con los requerimientos

Cumplir con las expectativas del cliente.

2.14.1. Normas de Calidad Internacionales (ISO)

ISO. Es la denominación con que se conoce a la International Organization for Standardization (IOS); sin embargo, considerando la tendencia a la estandarización global - homogeneización - que propone dicha organización, es que se le asigna la sigla ISO, vocablo que proviene del griego "iso" que en castellano significa "igual".

2.14.2. ISO 9126

La calidad de producto queda definida a un alto nivel de abstracción por las características denominadas *usabilidad, funcionalidad, confiabilidad, eficiencia, Mantenibilidad y portabilidad*. Luego, se emplea un proceso de descomposición recursivo basado en sub características y atributos, en consideración de una meta de evaluación y perfil de usuario dados. No obstante, nuevos desafíos surgen al evaluar la calidad en uso, como discutiremos en las siguientes secciones. Cabe destacar que hemos considerado importante la incorporación de un marco conceptual de calidad que sirva de guía a los evaluadores en el proceso de clasificación de entes, modelos y métricas asociadas.

Los aspectos más relevantes del marco conceptual de calidad, sin discutir los detalles en profundidad. Se intenta capturar a un alto nivel de abstracción las principales relaciones entre los factores de calidad en consideración de los entes que podrían intervenir en el proceso de aseguramiento de calidad. En este esquema implícitamente observamos que cada factor de calidad pertenece a un ente del dominio o mundo real, tangible o intangible. Debido a que un ente sólo puede ser medido a través de sus atributos, es preciso definir métricas de los atributos del ente e instrumentos adecuados, como cuestionarios, para poder analizar y conjeturar a partir de los números. Además, cada factor de calidad puede ser descrito por un modelo de calidad, como es el caso de la norma **ISO 9126-1** para calidad de producto.

2.14.3. Modelos, Métricas e Instrumentos para Medir Calidad en Uso.

La calidad en uso está directamente condicionada por la percepción que el usuario tiene del producto en uso, en un contexto determinado. Por lo que nuestra propuesta (surgida de las ideas plasmadas en el nuevo modelo de calidad de **ISO**), indica que debemos medir la calidad en uso por medio de características de alto nivel como efectividad, productividad, seguridad y satisfacción del usuario, en contextos o escenarios específicos de uso. Pero una vez más, estas características son de tal nivel de abstracción que no es directamente cuantificables, por lo que necesitaremos definir

atributos que sí lo sean. Ya que el usuario está directamente involucrado en la obtención de la medida de esta calidad, es decir estará involucrado en el proceso de evaluación, se debe crear herramientas participativas, donde él mismo, tal vez con la intervención de observadores como agentes automáticos o personas expertas. En el desarrollo del proceso de evaluación de la calidad en uso se deberá considerar los siguientes pasos principales:

- Análisis de los diferentes contextos y tareas en que al producto en uso ha de ser sometido a prueba conforme a la meta de la evaluación; en este punto se definirán los diferentes perfiles de usuarios adaptados a cada contexto y tarea.
- Especificación de los requerimientos de calidad en uso para los contextos y tareas identificados, en los que se ha de situar a los usuarios (evaluador-evaluado) para cada perfil. Se establecerá el modelo de calidad en uso, a partir de una definición y especificación precisa de las características y atributos.
- A partir del modelo de calidad especificado, se diseñarán los criterios, métricas, instrumentos como cuestionarios, casos de prueba, herramientas y procedimientos de evaluación.
- Una vez planificado y diseñado el proceso de evaluación se llevará a fase de implementación.
- Por último, se establecerán las conclusiones y recomendaciones. Como comentario de algunos pasos podemos decir que la evaluación deberá establecer y especificar el contexto de uso y las tareas que deberán realizarse, el número de usuarios participantes necesarias para representar a cada uno de los grupos definidos (generalmente no más de cinco por perfil. Respecto al modelo de calidad en uso, tomando como base al establecido en **ISO 9126-1**, las características definidas son *efectividad*, *productividad*, *seguridad* y *satisfacción de usuario*. A modo de ejemplo, para la característica de *efectividad* se pueden especificar métricas como:

- Nivel de completitud de la tarea. Si se selecciona una tarea específica, de la cual se conoce el resultado a obtener una vez realizada, se mide el nivel de logro en completar la tarea correctamente.
- Media de objetivos alcanzados, es decir, la media de las tareas realizadas correctamente. Algunos de los atributos que van a influir a la característica de *productividad* y que podemos medir son:
- Tiempo consumido en la tarea: definido como el tiempo que ha tardado un usuario en realizar una tarea previamente establecida. Se puede obtener el tiempo medio consumido para un tipo de usuario y compararlo con el que hubiera tardado un grupo de expertos.
- Eficiencia en la completitud: definida como el cociente entre el nivel de completitud y el tiempo medio consumido en la tarea. - Eficiencia de objetivos alcanzados: definida como el cociente entre la media de los objetivos alcanzados y el promedio de los tiempos medios consumidos.
-

Para medir la *satisfacción del usuario* se utilizarán cuestionarios, compuestos de 30 preguntas aproximadamente, en el que el objetivo sea considerar aspectos como la apariencia estética, la velocidad percibida, la relevancia de contenidos, si las funciones son adecuadas a la funcionalidad esperada, entre otros. Podemos tomar como ejemplos el cuestionario WebQual o cuestionarios de satisfacción como SUMI / WAMMI. Tras el proceso de definición de las métricas e instrumentos debemos proceder a su implementación, para ello utilizaremos técnicas de ciclo de vida tardía donde el sistema o producto ya está en uso, para las que se contará con encuestas y cuestionarios además de la observación del comportamiento del usuario respecto al sistema dentro de un laboratorio de prueba de "usabilidad". Esta prueba o test deberá ser realizado en un ambiente cercano, en la medida de lo posible, al de uso habitual del producto para el perfil de usuario-evaluador. Como hemos mencionado anteriormente también se pueden incorporar agentes automáticos, que favorecerán al

proceso de observación y comprobación de las acciones del usuario, evitando en lo posible la influencia que un observador humano podría introducir en el proceso.

2.15. Pruebas de Calidad de Software

Las pruebas de software que existen, básicamente, se pueden agrupar en dos grupos: las pruebas funcionales y las pruebas no funcionales. Sin embargo, hay más tipos de pruebas, por ejemplo, pruebas unitarias, pruebas de integración o pruebas de aceptación, pero estos tipos se pueden agrupar dentro de los dos grupos anteriores.

2.15.1. Tipos de Pruebas

Dentro de las pruebas funcionales tenemos:

- Pruebas unitarias.
- Pruebas de aceptación.
- Pruebas de integración.
- Pruebas de regresión.

Las pruebas no funcionales son:

- Pruebas de carga.
- Pruebas de estrés.
- Pruebas de escalabilidad.
- Pruebas de portabilidad.

Las pruebas de caja negra, no son tipos de pruebas, sino técnicas de pruebas de software, que serán aplicadas en este proyecto.

2.16. Estimación de Costos de Desarrollo de Software

2.16.1. Modelo COCOMO.

CONstructive COSt MESel del (COCOMO) es un algorítmico Modelo de la valoración del coste del software convertido cerca Barry Boehm. El modelo utiliza una

básica regresión fórmula, con los parámetros que se derivan de datos históricos del proyecto y de características actuales del proyecto.

Fue publicado en 1981 Barry J. Boehm Libro Economía de la tecnología de dotación lógica como modelo para estimar esfuerzo, coste, y el horario para los proyectos del software. Dibujó en un estudio de 63 proyectos en TRW Espacio aéreo donde Barry Boehm era el director de la investigación y de la tecnología del software en 1981. El estudio examinó los proyectos que se extendían de tamaño a partir de 2000 a 100.000 líneas del código, y lenguajes de programación que se extienden de ensambla a PL/I. Estos proyectos fueron basados en modelo de la cascada del desarrollo del software que era el proceso frecuente del desarrollo del software en 1981.

El modelo COCOMO II consta de 3 ecuaciones las que mostraremos a continuación:

$$E = a(KLDC)^b * m(X), \text{ en personas mes}$$

Ecuación 1

$$D = c(E)^d, \text{ en meses}$$

Ecuación 2

$$P = \frac{E}{D}, \text{ en personas}$$

Ecuación 3

Dónde:

E: Esfuerzo requerido por el proyecto expresado en persona-mes.

D: Tiempo requerido por el proyecto expresado en meses.

P: Número de personas requeridas para el proyecto.

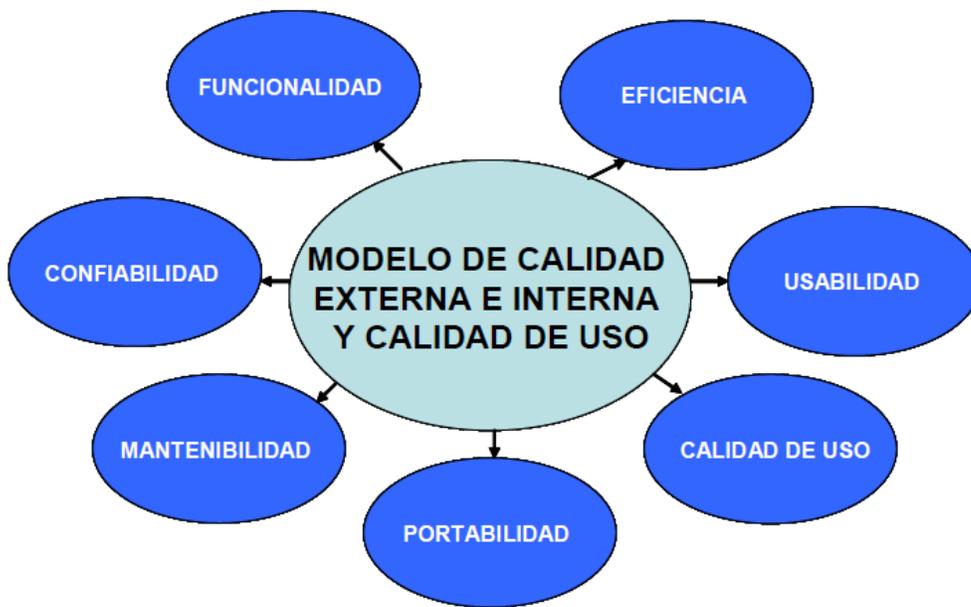
a, b, c y d: Constantes con valores definidos según cada sub-modelo.

KLDC: Cantidad de líneas de código distribuidas en miles.

A la vez cada modelo se subdivide en modos, los mismos son:

- **Modo orgánico:** es un pequeño grupo de programadores experimentados desarrollando proyectos de software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas docenas de miles (tamaño medio).
- **Modo semi – libre o semi – acoplado:** Corresponde a un esquema intermedio entre modo orgánico y el rígido, el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- **Modo rígido o empotrado:** El proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único, siendo difícil basarse en la experiencia puesto que puede no haberla.

Figura 2- 14 Modelo de Calidad



Fuente: (ISO & ISO, 2003)

Tabla 2- 7 Esfuerzo de Personas-mes

Proyecto del software	a_b	b_b	c_b	d_b
Orgánico	2.4	1.05	2.5	0.38
Semi-separado	3.0	1.12	2.5	0.35

Encajado	3.6	1.20	2.5	0.32
-----------------	-----	------	-----	------

Fuente: (Calero, 2010)

Cada uno de las 15 cualidades recibe un grado en una escala del seis-punto que se extiende de “muy bajo” a “superior” (en importancia o valor). Un multiplicador del esfuerzo de la tabla abajo se aplica al grado. El producto de todos los multiplicadores del esfuerzo da lugar a *coeficiente de adaptación del esfuerzo (EAF)*. Los valores típicos para EAF se extienden a partir de la 0.9 a 1.4.

Tabla 2- 8 Tabla de Coste de Software

Conductores del coste	Grados					
	Muy bajo	Bajo	Nominal	Alto	Muy arriba	Superior
Cualidades de producto						
Confiabilidad requerida del software	0.75	0.88	1.00	1.15	1.40	
Tamaño de la base de datos del uso		0.94	1.00	1.08	1.16	
Complejidad del producto	0.70	0.85	1.00	1.15	1.30	1.65
Cualidades del hardware						

Apremios de funcionamiento Run-time			1.00	1.11	1.30	1.66
Apremios de la memoria			1.00	1.06	1.21	1.56
Volatilidad del ambiente virtual de la máquina	0.87		1.00	1.15	1.30	
Tiempo de turnabout requerido	0.87		1.00	1.07	1.15	
Cualidades del personal						
Capacidad del analista	1.46	1.19	1.00	0.86	0.71	
Experiencia de los usos	1.29	1.13	1.00	0.91	0.82	
Capacidad de la Software Engineer	1.42	1.17	1.00	0.86	0.70	
Experiencia virtual de la máquina	1.21	1.10	1.00	0.90		
Experiencia del lenguaje de programación	1.14	1.07	1.00	0.95		
Cualidades del proyecto						
Uso de las herramientas del software	1.24	1.10	1.00	0.91	0.82	
Uso de los métodos de la tecnología de dotación lógica	1.24	1.10	1.00	0.91	0.83	

Horario requerido del desarrollo 1.23 1.08 1.00 1.04 1.10

Fuente: (Calero, 2010)

Donde está el esfuerzo E aplicado en persona-meses, KLoC es el número estimado de millares de líneas entregadas de código para el proyecto, yEAF es el factor calculado arriba. El coeficiente a_i y el exponente b_i se dan en la tabla siguiente.

Tabla 2- 9 Esfuerzo Aplicado en Líneas de Código

Proyecto del software	a_i	b_i
Orgánico	3.2	1.05
Semi-separado	3.0	1.12
Encajado	2.8	1.20

Fuente: (Calero, 2010)

El tiempo de desarrollo D aplicaciones del cálculo E de la misma forma que en el COCOMO básico.

Marco Aplicativo

3.1. Análisis de la Situación Actual

ADEFUSA Bolivia, una entidad sin fines de lucro, cuya misión es apoyar a emprendimientos deportivos a nivel local, departamental y nacional. Este trabajo se realiza mediante programas y proyectos con visión a la promoción en niveles de recreación, formación y competencia en diferentes disciplinas deportivas. Caso concreto, en la disciplina de fútbol, viene trabajando desde los inicios de la conformación de la fundación.

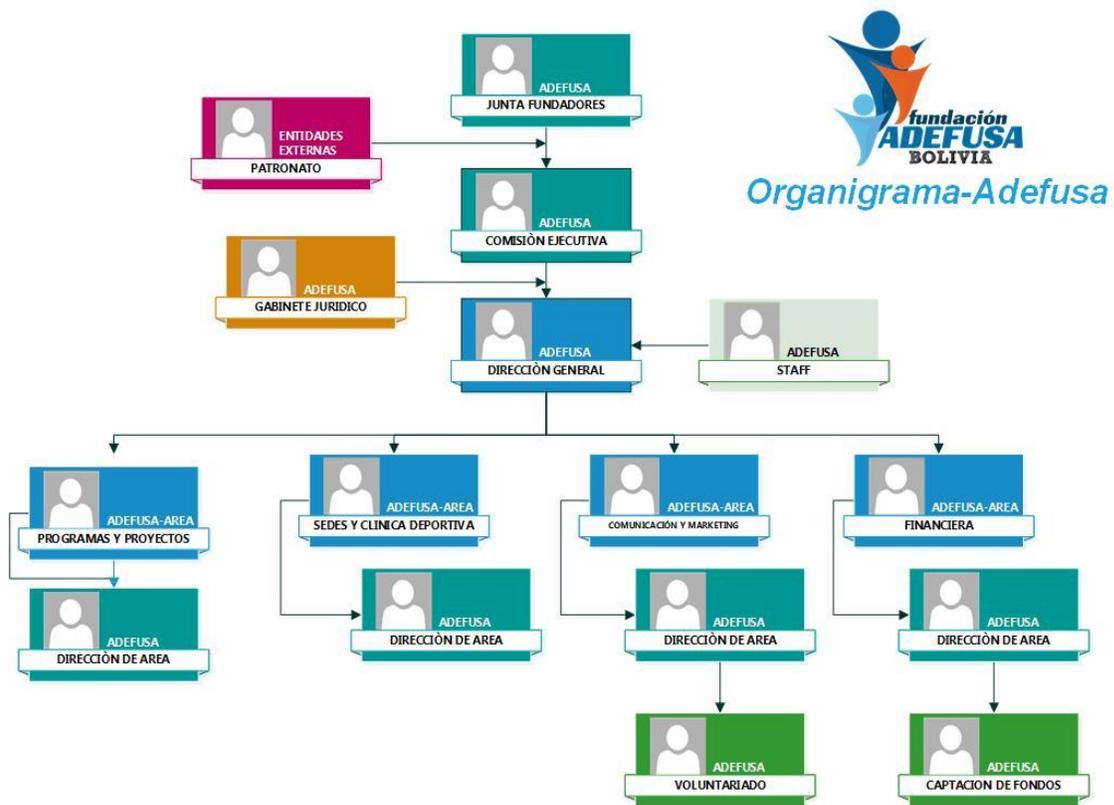
La fundación cuenta con diferentes escuelas de fútbol, situadas en diferentes zonas de la ciudad del El Alto, vienen trabajando con la niñez y juventud alteña ejecutando proyectos deportivos como ICOTA (Identificación, Cooperación para el Talento Alteño), cuyo objetivo principal es la de trabajar con talentos mejorando la técnica y táctica del Jugador.

También cuenta con el proyecto Escuelas de Fútbol, Escuelas de Vida, cuya colaboración de Ricerca E Cooperazione (fundación italiana) actualmente como APEA, por una Educación Activa, son los brazos operativos y acción colaborativa con niños y jóvenes de escasos recursos económicos y/o familias disgregadas, otorgando una oportunidad de vida mediante el fútbol.

3.2. Organigrama Institucional

La organización de la institución está compuesta por Dirección y áreas como Área de Programas y Proyectos, Sedes y Clínica Deportiva, Comunicación y Marketing y Finanzas. Como se muestra en el gráfico siguiente.

Figura 3- 1 Organigrama Adefusa



Fuente: Elaboración Propia

3.3. Escuelas de Fútbol Adefusa

Las escuelas deportivas de Fútbol Adefusa, en su interior cuenta con su propia organización institucional, contribuyendo al mejor desempeño y flujo de trabajo e información.

Coordinador de Escuelas Deportivas. Comprende un coordinador de Escuelas por disciplina deportiva. Tal es el caso de las Escuelas deportivas de fútbol.

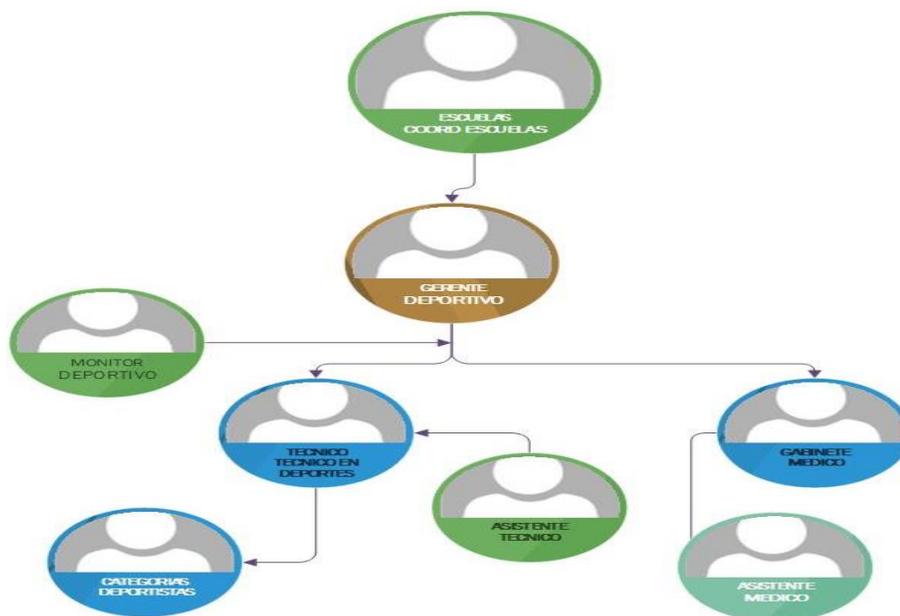
Gerente Deportivo. Cada escuela de futbol está a cargo por un Profesional Entrenador Deportivo quien es el encargado de velar por el funcionamiento continuo y la ejecución del plan de anual de entrenamiento y sesiones planificadas para cada jugador.

Monitor Deportivo. Es encargado de monitorear, controlar, vigilar las actividades dentro la escuela deportiva. Además, que coadyuva en la labor de informar, inscribir, controlar la asistencia de deportistas.

Entrenador Técnico de futbol. Es el encargado de ejecutar las sesiones de futbol, de acuerdo al plan de entrenamiento anual preparado. Éste entrenador puede tener Selecciones, Formación y Aficionados por categorías.

Asistente Técnico. Es voluntario quien coadyuva a la labor del Entrenador Técnico de futbol, en el cumplimiento de la labor diaria de entrenamiento.

Figura 3- 2 Organigrama Escuela Deportiva



Fuente: Elaboración Propia

3.4. Ciclo de Vida del Software

Para el estudio de caso, se ve por conveniente aplicar el **modelo incremental**, es el que permite que las fases de análisis, diseño, desarrollo y pruebas se retroalimentan continuamente, y que empiecen lo antes posible. Permitirá atender a posibles cambios a las necesidades del usuario, nuevas herramientas o componentes que los desarrolladores descubran y que faciliten el diseño y/o proporcionen nuevas funcionalidades.

3.5. Fases de la Metodología UWE.

3.5.1. Captura, Análisis y Especificación de Requisitos.

La entidad va siguiendo y ejecutando un conjunto de actividades que se encuentran ya enmarcadas como proceso de trabajo propio de la misma. Se comienza con la obtención de requisitos, para obtener información necesaria, para ello se realizó las siguientes técnicas de investigación:

Tabla 3- 1 Técnica de recopilación de datos

TÉCNICA	DESCRIPCIÓN
Observación	Indirecta, porque se realiza la investigación mediante preguntas al sujeto (personal) involucrado, en su desarrollo normal de la actividad encomendada de la entidad.
Entrevista	De forma abierta, buscando un tema en específico, de acuerdo al rol que cumple en la entidad. Buscando la información sobre su forma de trabajo y la opinión que tiene sobre ella.
Revisión Documental	Se realiza esta actividad, para tener más detalle de la generación de información de la entidad específica con respecto al caso de estudio.

Requerimientos Funcionales. A partir de la información obtenida en la obtención del requerimiento de software se define los siguientes requisitos funcionales.

Tabla 3- 2 Lista de Requerimientos Funcionales

Nro.	REQUERIMIENTO	CASO DE USO	ACTOR	CLASIFICACIÓN
				FURPS PLUS
RF1	El Sistema debe permitir registrar nuevas creaciones de escuelas deportivas.	Registrar Escuela Deportiva (añadir, modificar)	Secretaria	F
RF2	El Sistema debe permitir dar de alta y mantener a una escuela deportiva	Mantener Escuela (añadir, ecualizar, eliminar).	Coordinador Escuelas Deportivas	F
RF3	El Sistema debe permitir registrar y dar de alta a nuevo personal (entrenador, asistente técnico).	Mantener personal (añadir, actualizar, eliminar).	Coordinador Escuelas Deportivas	F
RF4	El Sistema de permitir realizar búsquedas de entrenadores y asistentes técnicos con horarios designados.	Reportar estado personal	Coordinador Escuelas Deportivas	F
RF5	El Sistema debe permitir designar a un entrenador y su asistente, a una escuela que este dado de alta para el trabajo con los deportistas.	Designar Personal (añadir, actualizar).	Coordinador Escuelas Deportivas	F
RF6	El Sistema debe registrar nuevos deportistas.	Mantener Deportista (añadir, actualizar)	Secretaria, Monitor	F
RF7	El Sistema debe permitir asignar o reasignar a una escuela deportiva con sus horarios de entrenamiento al deportista.	Designar Escuela deportiva (añadir, actualizar).	Secretaria, Monitor	F
RF8	El Sistema debe permitir al deportista revisar sus datos personales y modificarlos	Datos Deportista (ver, actualizar)	Deportista	F
RF9	El Sistema debe permitir crear y registrar un plan de entrenamiento anual	Mantener Plan de Entrenamiento	Entrenador	F

		(añadir, actualizar, eliminar)		
RF10	El Sistema debe permitir crear y registrar sesiones de Entrenamiento	Mantener Sesiones de Entrenamiento (añadir, actualizar, eliminar)	Entrenador	F
RF11	El Sistema debe generar informe técnico del plan de entrenamiento planificado	Generar Informe Técnico	Entrenador	F
RF12	El sistema debe permitir registrar el expediente del deportista las cuales contiene datos de ficha técnica	Mantener expediente deportista (añadir, actualizar)	Entrenador, Monitor	F
RF13	El Sistema debe permitir ingresar datos de la Evaluación Técnica Realizado y Actualizar la ficha técnica del Deportista.	Mantener Ficha Técnica del deportista (actualizar)	Entrenador, Monitor	F
RF14	El Sistema debe permitir generar Informe Técnico con la evaluación Técnica.	Generar Informe Evaluación Técnica	Entrenador	F

Requerimientos no Funcionales. De acuerdo a la característica y criterios del producto (sistema), los requisitos no funcionales se detallan a continuación.

Tabla 3- 3 Lista de Requerimientos no Funcionales

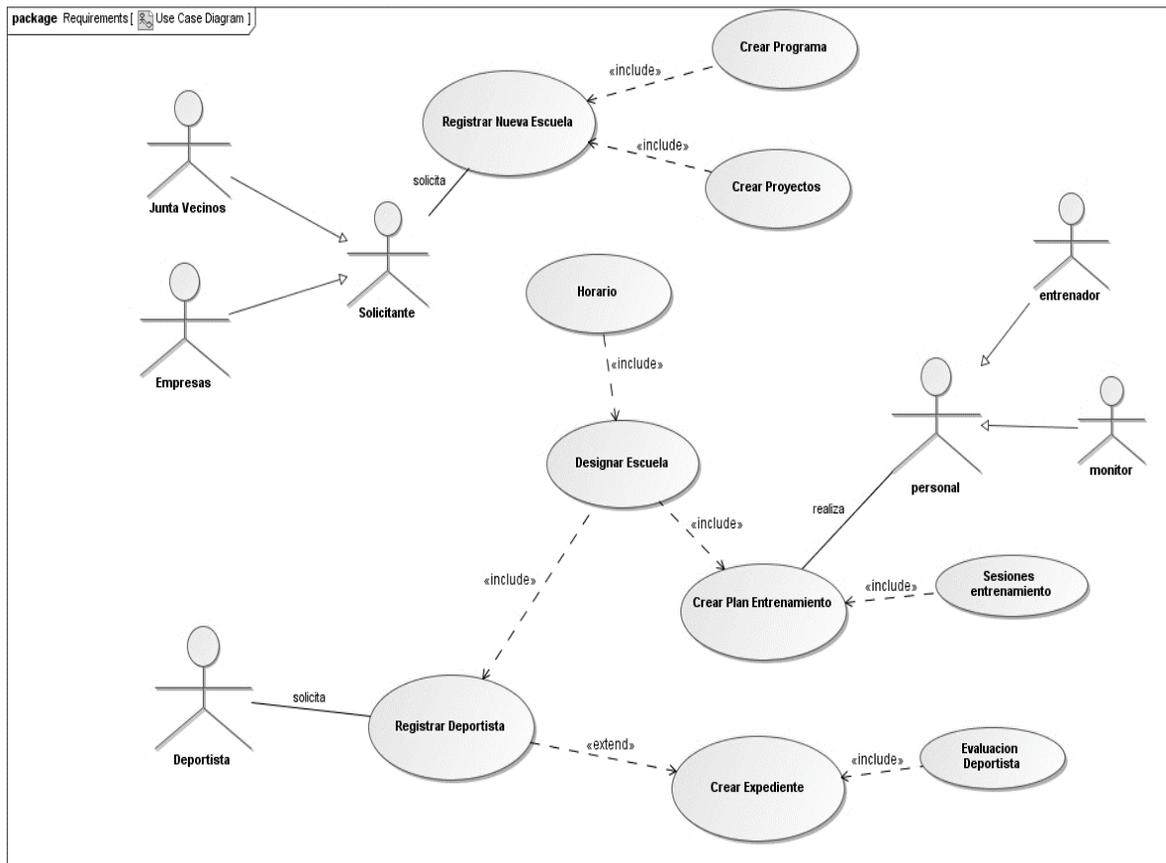
Nro.	REQUERIMIENTO	CLASIFICACIÓN
		FURPS PLUS
RNF1	La funcionalidad del sistema y transacción de negocio debe responder al usuario en menor tiempo requerido (máximo en 5 segundos).	P
RNF2	El sistema debe ser capaz de operar adecuadamente con usuarios y sesiones concurrentes.	p
RNF3	Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.	R
RNF4	Todos los datos del sistema deben respaldarse en tiempo máximo de 48 horas. Los respaldos deben ser	R

	almacenados en una localidad segura ubicada en hardware distinto al que reside el sistema.	
RNF5	El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 4 horas.	U
RNF6	La tasa de errores cometidos por el usuario deberá ser menor del 1% de las transacciones totales ejecutadas en el sistema.	U
RNF7	El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.	U
RNF8	El sistema debe tener una disponibilidad del 99,99% de las veces en que un usuario intente accederlo.	PLUS

3.5.2. Diseño del Sistema.

Diagrama de Caso de Uso. De acuerdo al análisis de requerimientos realizado se pudo diseñar el siguiente Caso de Uso de negocio, reflejando los actores que interactúan con el negocio.

Figura 3- 3 Diagrama de caso uso del negocio

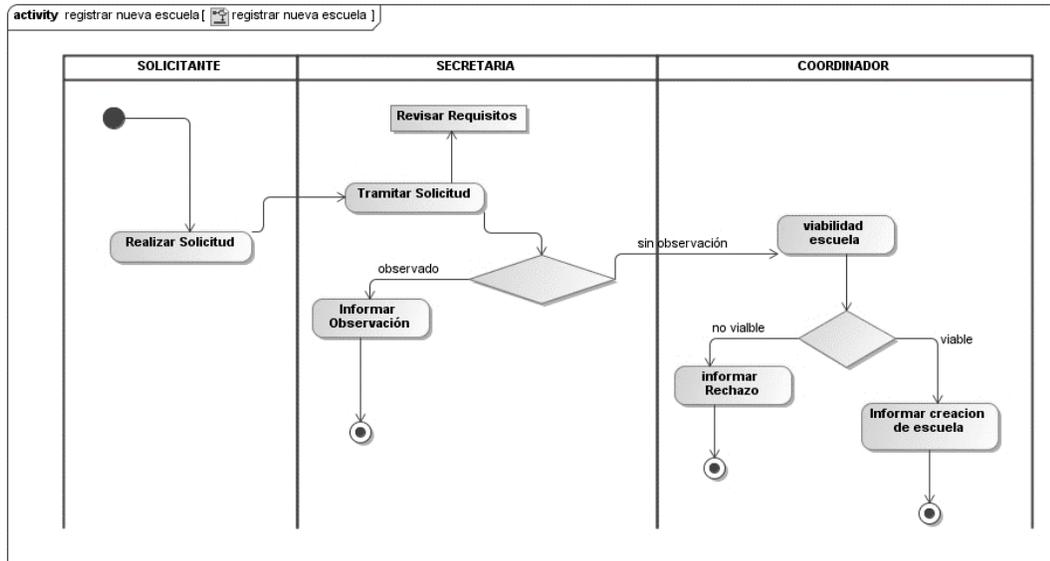


Fuente: Elaboracion propia

Diagrama de actividad. De acuerdo al caso de uso del negocio (CUN) de la institución, se detallan los siguientes, diagramas de actividades.

- Entidad Registrar nueva escuela.

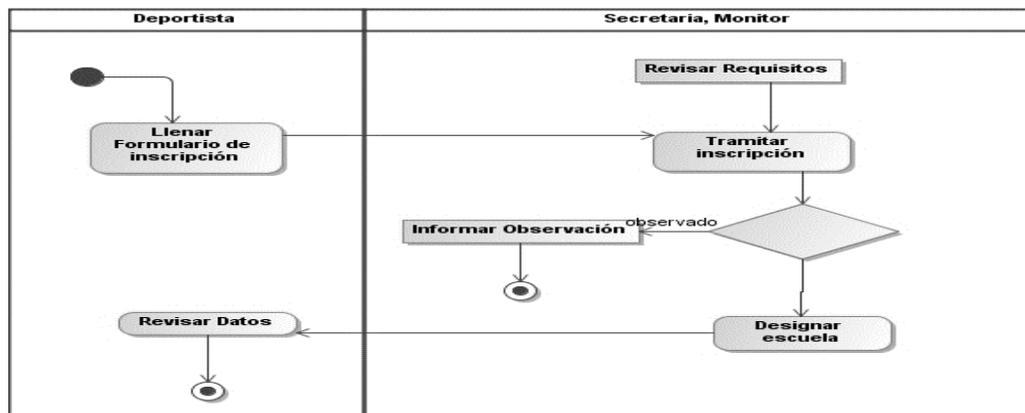
Figura 3- 4 Registrara Escuela



Fuente: Elaboracion propia

- Registrar Deportista.

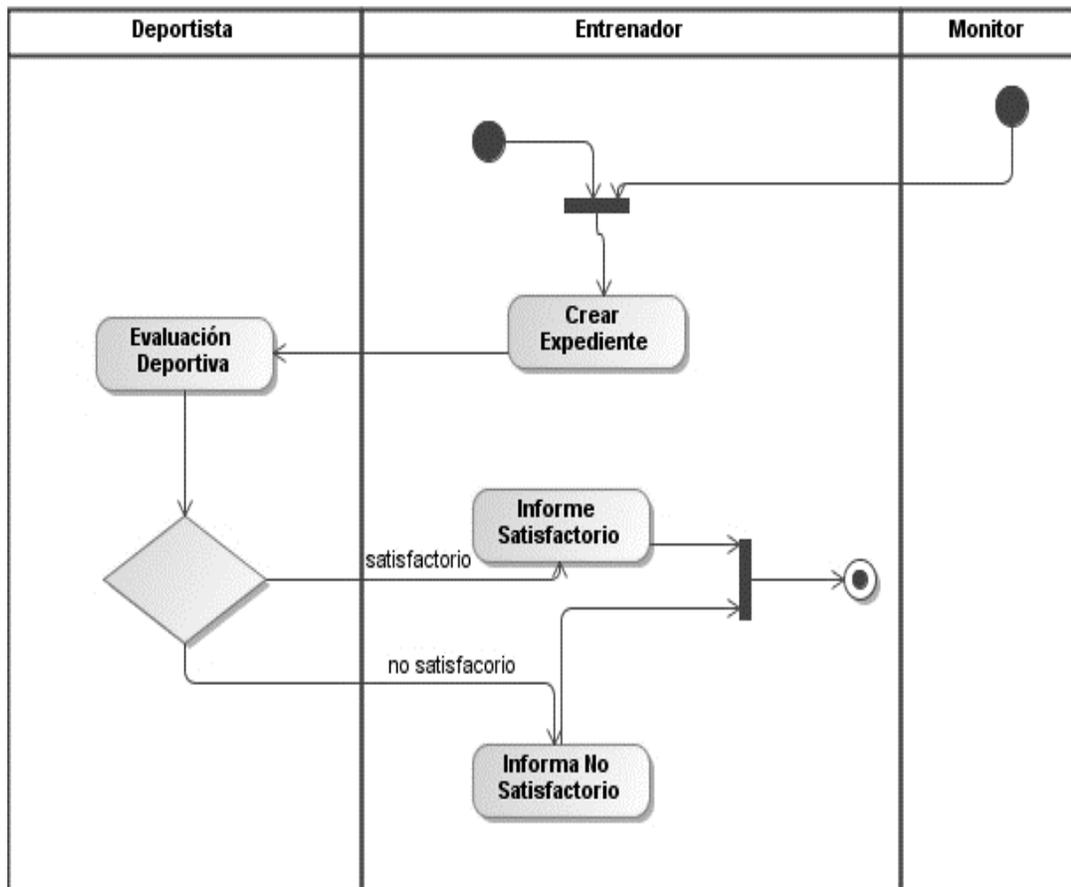
Figura 3- 5 Registrar deportista



Fuente: Elaboracion propia

- Crear expediente.

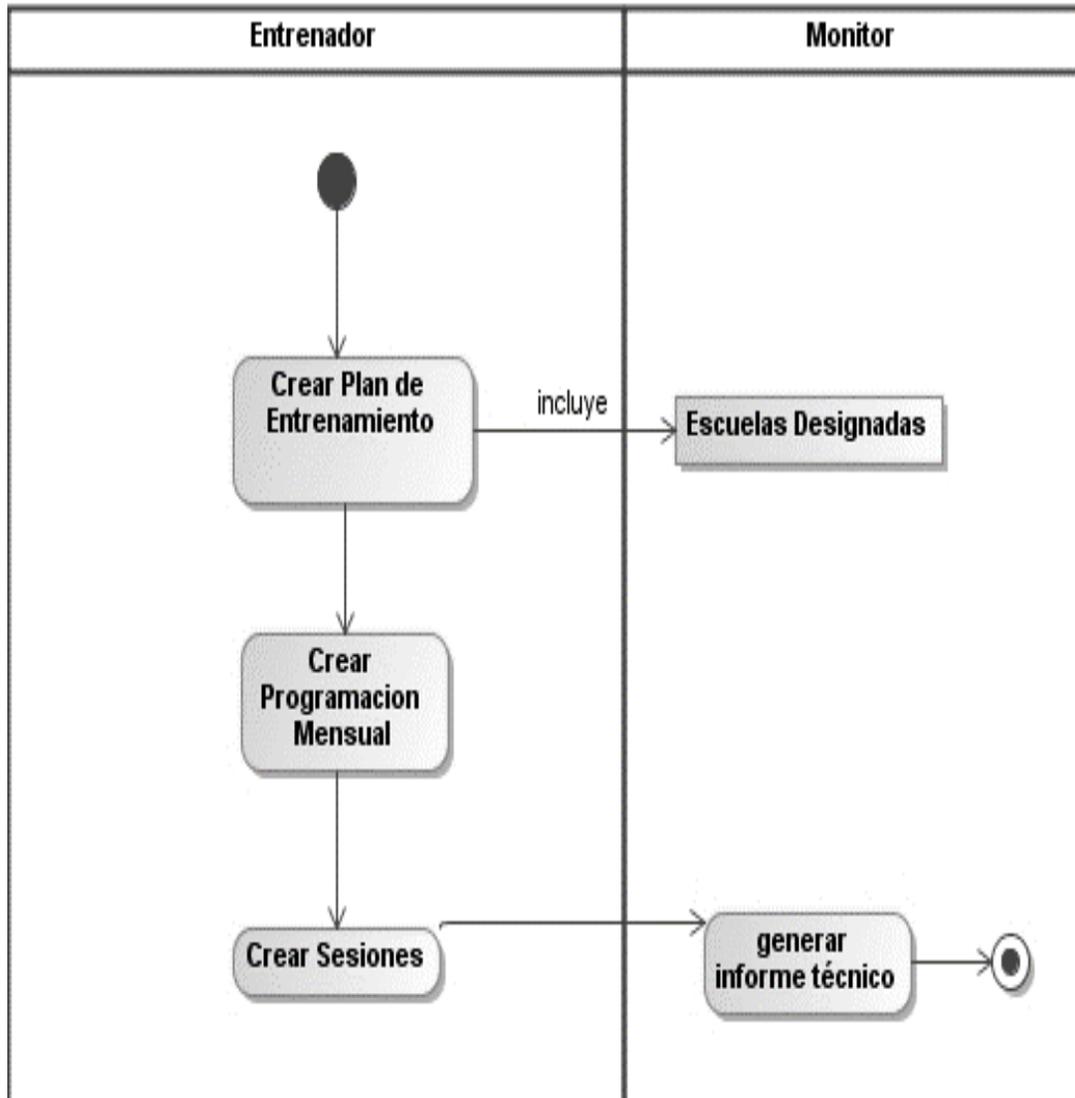
Figura 3- 6 Crear Expediente deportista



Fuente: Elaboracion propia

- Crear plan de entrenamiento

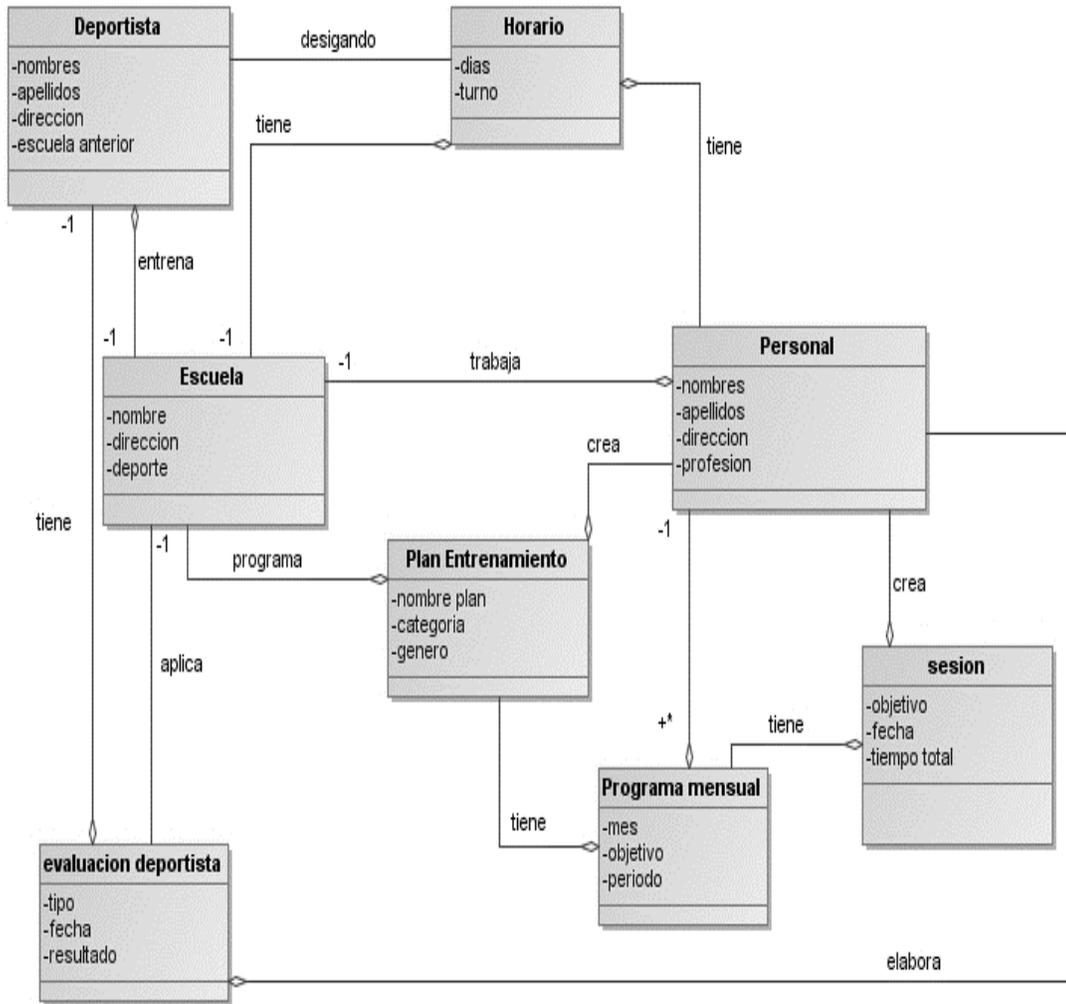
Figura 3- 7 Crear Plan de entrenamiento



Fuente: Elaboracion propia

Modelo conceptual

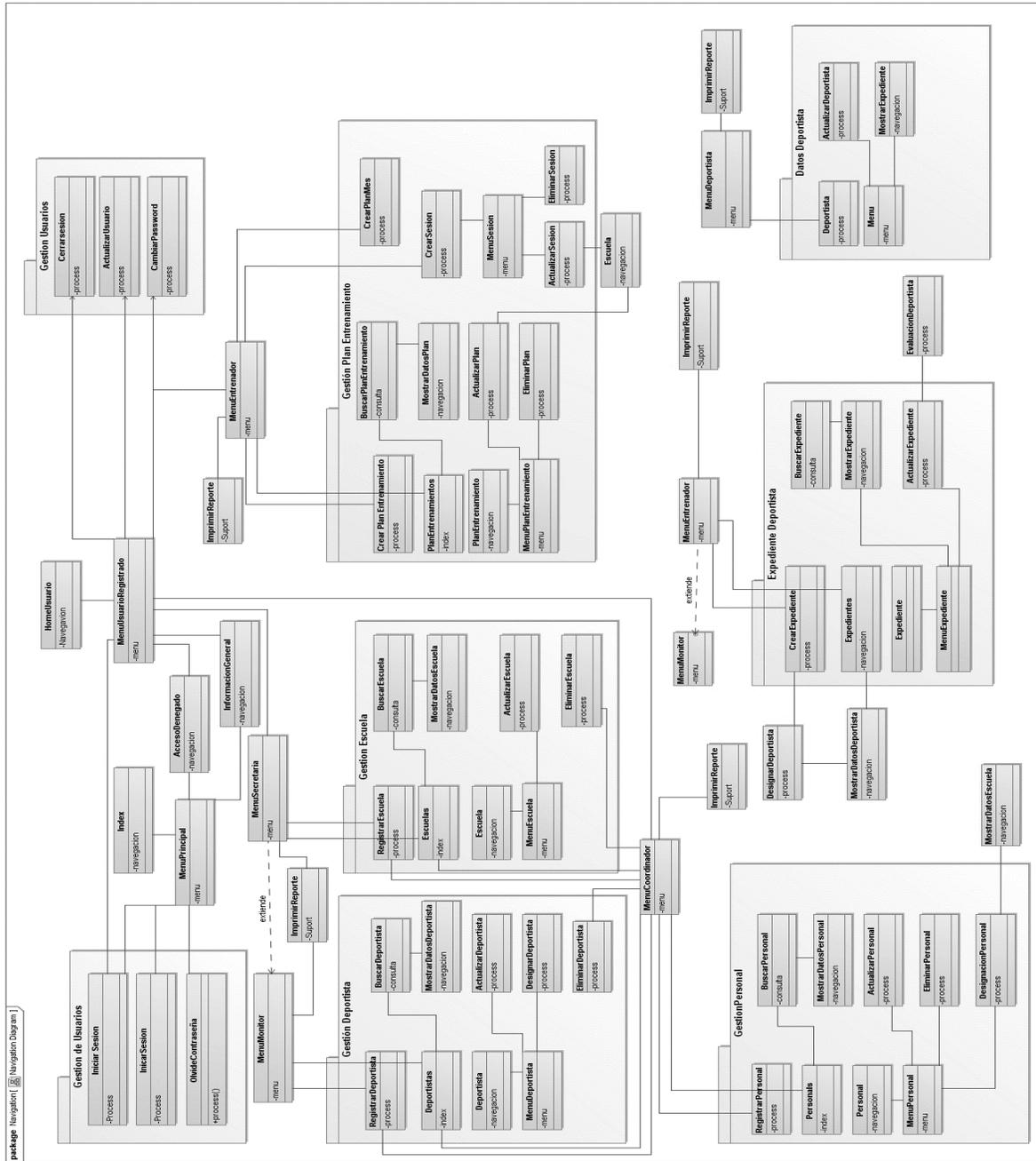
Figura 3- 8 Diagrama de clases



Fuente: Elaboracion propia

Modelo de navegación

Figura 3- 9 Navegación de sistema



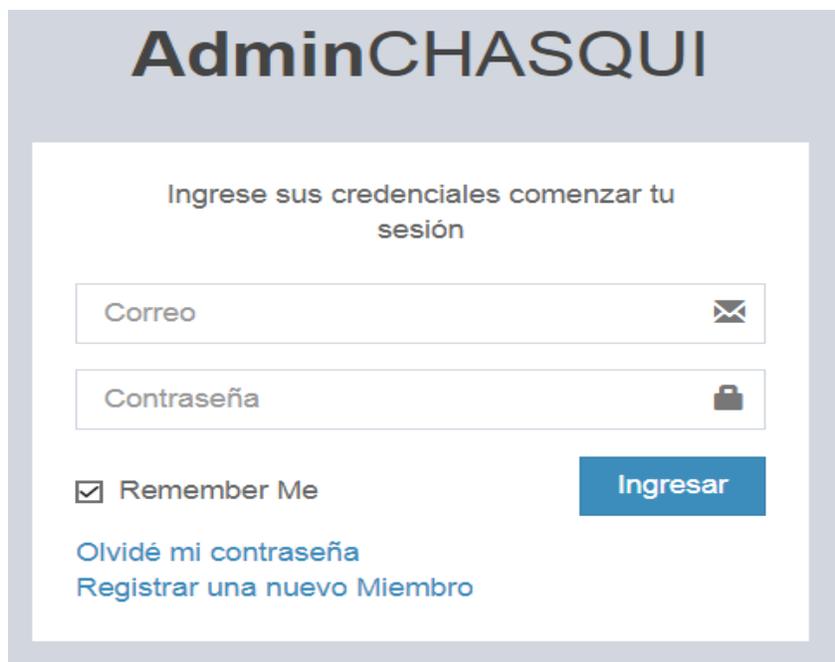
Fuente: Elaboración propia

3.5.3. Codificación de software.

En esta fase se realiza la programación de la Aplicación Web de acuerdo con la fase de diseño realizada anteriormente, además cumpliendo con los requerimientos presentados en la fase de captura, análisis y especificación de requisitos mediante las interfaces requeridas.

Ingreso de Credenciales. Al ingreso del sistema podemos observar la pantalla de Inicio de Sesión de la Aplicación Web, que de acuerdo al rol que tenga en la misma.

Figura 3- 10 inicio de sesión



AdminCHASQUI

Ingrese sus credenciales comenzar tu sesión

Correo

Contraseña

Remember Me

Ingresar

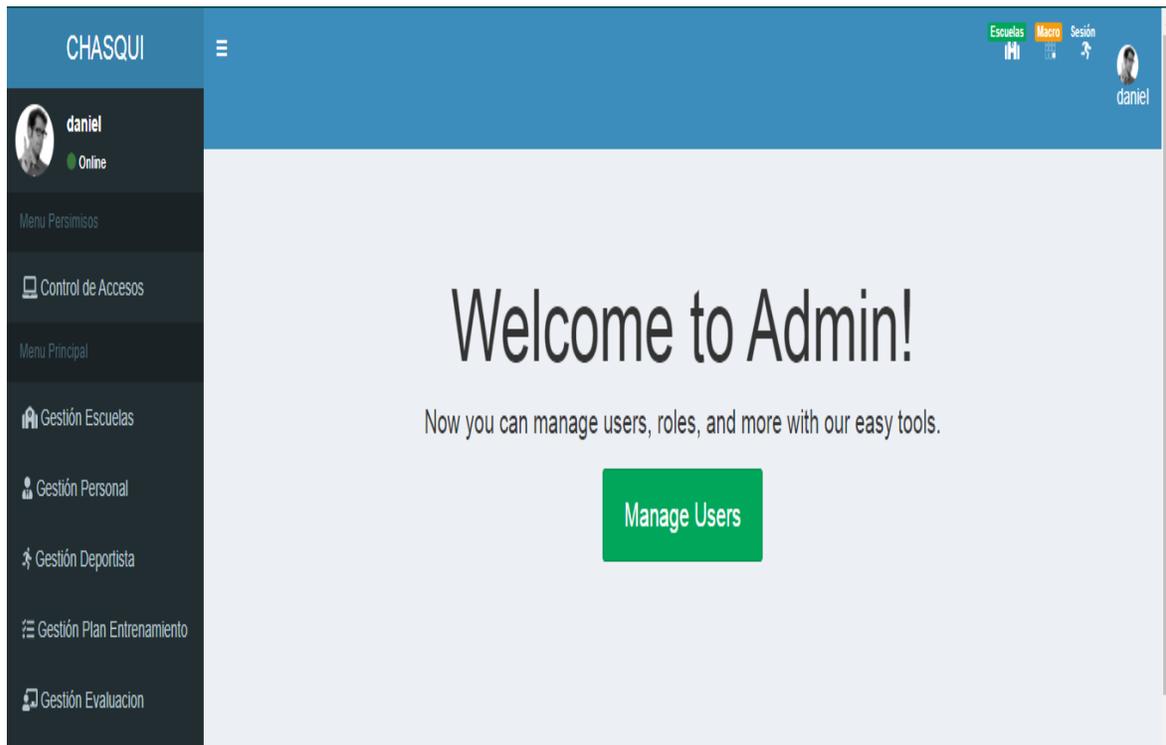
[Olvidé mi contraseña](#)

[Registrar una nuevo Miembro](#)

Fuente: Elaboracion propia

Presentación del Sistema. Es la pantalla principal, esto otorgando diferentes opciones de trabajo de acuerdo al rol otorgado mediante el sistema.

Figura 3- 11 pantalla de Bienvenida



Fuente: Elaboracion propia

Modulo Acceso de usuarios basado en roles (RBAC). Este módulo permite dar acceso al sistema de acuerdo al rol y responsabilidades que tiene el trabajador con la institución

Figura 3- 12 Control de Accesos

The screenshot displays a user management interface. On the left is a dark sidebar with a user profile for 'daniel' (Online) and a menu with options like 'Control de Accesos', 'Usuarios', 'Perfiles', 'Roles', 'Tipo-Usuarios', 'Estado', 'Level One', 'Gestión Escuelas', 'Gestión Personal', 'Gestión Deportista', and 'Gestión Plan Entrenamiento'. The main content area is titled 'Update User: 5' and contains the following form fields:

- Status Id:** A text input field containing 'Active'.
- Role Id:** A text input field containing 'UserAdmin'.
- User Type Id:** A dropdown menu with the following options: 'Please Choose One', 'Secretaria', 'Monitor' (highlighted in blue), 'Coordinador', 'Gerente', 'GerenteEjecutivo', and 'Entrenador'.

Below the dropdown, the email address 'pedromamani@pedro.com' is displayed. At the bottom of the form is a blue 'Update' button.

Fuente: Elaboracion propia

Módulo Gestión Escuelas Deportivas. Este modulo, se puede controlar todo con referente a las escuelas deportivas, las sedes conde se encuentran y la cantidad de escuelas y personal designado en la misma. Tambien puede filtrar por criterios de búsqueda.

Figura 3- 13 Escuelas Deportivas

The screenshot displays the 'Detalle Sede' (Venue Details) page in the CHASQUI system. The interface includes a dark sidebar on the left with the user profile 'daniel' (Online) and a menu with options like 'Gestión Escuelas', 'Cal. Campo Deportivo', and 'Sedes Deportivas'. The main content area features a blue header 'Detalle Sede' and a sub-header 'Zona Deportiva Huayna Potosi'. Below this are three action buttons: 'Actualizar' (blue), 'Eliminar' (red), and 'Listar' (green). A table provides details for the venue:

Nombre	Zona Deportiva Huayna Potosi
Tipo Terreno	CÉSPED CON CAMERINOS
Distrito	DM-5
Dirección	Av. Imperial Nro. 2025 lado Colegio Huayna Potosi

To the right of the table is a photo of a green sports field with a goal, and a green 'Actualizar Foto' button below it. At the bottom, a blue header 'Escuelas Deportiva' is followed by a table listing schools:

Total 1 elemento.

Escuela	
Escuela de Futbol Huayna Potosí Mañana	+ Nuevo  

Fuente: Elaboracion propia

Detalle de escuela con horario y personal designado a la escuela

Figura 3- 14 Detalle Escuela Deportiva

The screenshot displays a modal window titled 'Detalle Escuela' for 'Estadio Villa ingenio'. It includes a sidebar with a list of schools and a main content area with three sections: 'Actualizar', 'Eliminar', and 'Listar' buttons; a table of school details; a 'Horarios' section with a table of schedules; and a 'Personal Designado' section with a table of assigned staff.

Escuelas

Crear Escuela

Mostrando 1-2 de 2 elementos

#	Nombre Escuela
1	Escuela de Fútbol
2	Escuela de Fútbol

Detalle Escuela

Estadio Villa ingenio

Actualizar Eliminar Listar

Nombre Escuela	Escuela de Fútbol Villa ingenio Mañana
Sede	Estadio Villa ingenio
Deporte	Fútbol Sala

Horarios

Total 1 elemento.

Días	Turno	Hora Inicio	Hora Final	+ Nuevo
Martes, Jueves	M	08:00:00	11:30:00	

Personal Designado

Total 1 elemento.

Personal	Sede	Escuela	Horario	Turno
Juan Piter Sallinas	Estadio Villa ingenio	Escuela de Fútbol Villa ingenio Mañana	Martes, Jueves	M

Copyright © 2020 Adefu

Fuente: Elaboracion propia

Módulo Personal. Este módulo, controla al personal con referente datos personales, asignaciones de escuelas deportivas.

Figura 3- 15 Cargo Personal

Cargo Personal

Entrenador Profesional

Actualizar
Eliminar
Listar

Cargo Personal	Entrenador Profesional
Descripción del Cargo	Entrenador futbol base 01
Fecha Actualizada	2020-10-20

Cargo Personal

Total 1 elemento.

Personal	Ci	Direccion Actual	Fono-Cel	Email	Entidad Contratante	Foto
Daniel paye Aruquipa	4988154		70506082	danny@yahoo.es	GAMEA	

Fuente: Elaboracion propia

Lista y detalle del personal registrado en el sistema

Figura 3- 16 Detalle Personal

Personal								
Nuevo Personal								
Mostrando 1-6 de 6 elementos.								
#	Personal	Ci	Direccion Actual	Fono-Cel	Email	Entidad Contratante	Cargo	Foto
1	juan mamani mamani	74747474		75757575	juanmamani@yahoo.es	GAMEA	Entrenador Técnico	   
2	Rocio Gladis Mamani Mamani	7032479	av. ballivian nro. 55 zona Villa Juliana	73039571	rociomamani@yahoo.es	ESCUELA DE FÚTBOL RAMIRO CASTILLO	Monitor Deportivo	   
3	Ernesto Mamani Mamani	705064128		740511245	ernestomamani@yahoo.es	ESCUELA DE FÚTBOL RAMIRO CASTILLO	Entrenador Técnico	   
4	esteban paredes nomtaño	457842		68841251	estebanparedes@adefusabolivia.org	ADEFUSA	Monitor Deportivo	   
5	juan piter sallinas	49881542		74156248	juanpiter@telmo.net	MAYAS FC	Entrenador Técnico	   

Fuente: Elaboracion propia

Figura 3- 17 Información personal

Personal

49881542

[Actualizar](#)
[Eliminar](#)
[Listar](#)

Personal: juan piter sallinas Ci: 49881542 Genero: Masculino Cargo: Entrenador Técnico Entidad Contratante: MAYAS FC Direccion Actual: Fono-Cel: 74156248 Email: juanpiter@telmo.net	 Actualizar Foto
---	--

Disciplina Deportiva

Total 1 elemento.

Deporte
Futbol Sala

Escuelas Designada

Total 1 elemento.

Sede	Escuela	Horario	Turno	Nuevo
Estadio Villa ingenio	Escuela de Futsal Villa ingenio Mañana	Martes, Jueves	M	 

Plan Entrenamientos

Fuente: Elaboracion propia

Módulo Deportista. Este módulo, presenta el detalle del deportista inscrito con los datos de, escuela y deporte inscrita, las sesiones programadas y asistencia a la misma, evaluaciones realizadas.

Figura 3- 18 información Deportista

Deportista	Enrique Blanco Cespedes
Ci	100026536
Fecha Nacimiento	2000-06-21
Genero	Masculino
Direccion	la que sea nro 45a
Fono1	784512596
Email	enrique@yahoo.es
Intitucion Educativa	Colegio Los Andes
Grado Instuccion	4to primaria
Edad Actual	20



[Actualizar Foto](#)

Disciplina Deportiva

Total 1 elemento.

Escuela	Deporte	Horario	Entrenador	+ Nuevo
Escuela de Futbol Huayna Potosí Mañana	5	Lunes, Miercoles, Viernes	juan mamani mamani	 

Sesiones Programadas

Número Sesión	Deportista	Fecha	Asistencia	Escuela
No se encontraron resultados.				

Fuente: Elaboracion propia

Módulo Plan de Entrenamiento. Módulo, pensado para documentar y coadyuvar al entrenador a cargo de una categoría, con la planificación de periodos, meso ciclos, ciclos semana y sesiones de entrenamiento.

Figura 3- 19 Plan Anual de Entrenamiento

Plan Anual

Campeonato AFEAL 2020

exportar
Listar

Nro. Macro	1
Objetivo	Campeonato AFEAL 2020
Categoría	Infantil
Genero	Masculino
Escuela	Escuela de Futbol Huayna Potosí Mañana
Personal	juan mamani mamani
Obs	TENEMOS A CARGO 15 DEOPRTISTAS CON CIERTA TRAYECTORIA-
Created At	Oct 21, 2020

Periodos
Mesociclo
Micro - Ciclos
Fecha - Sesiones
Sesiones - Programadas

Total 2 elementos.

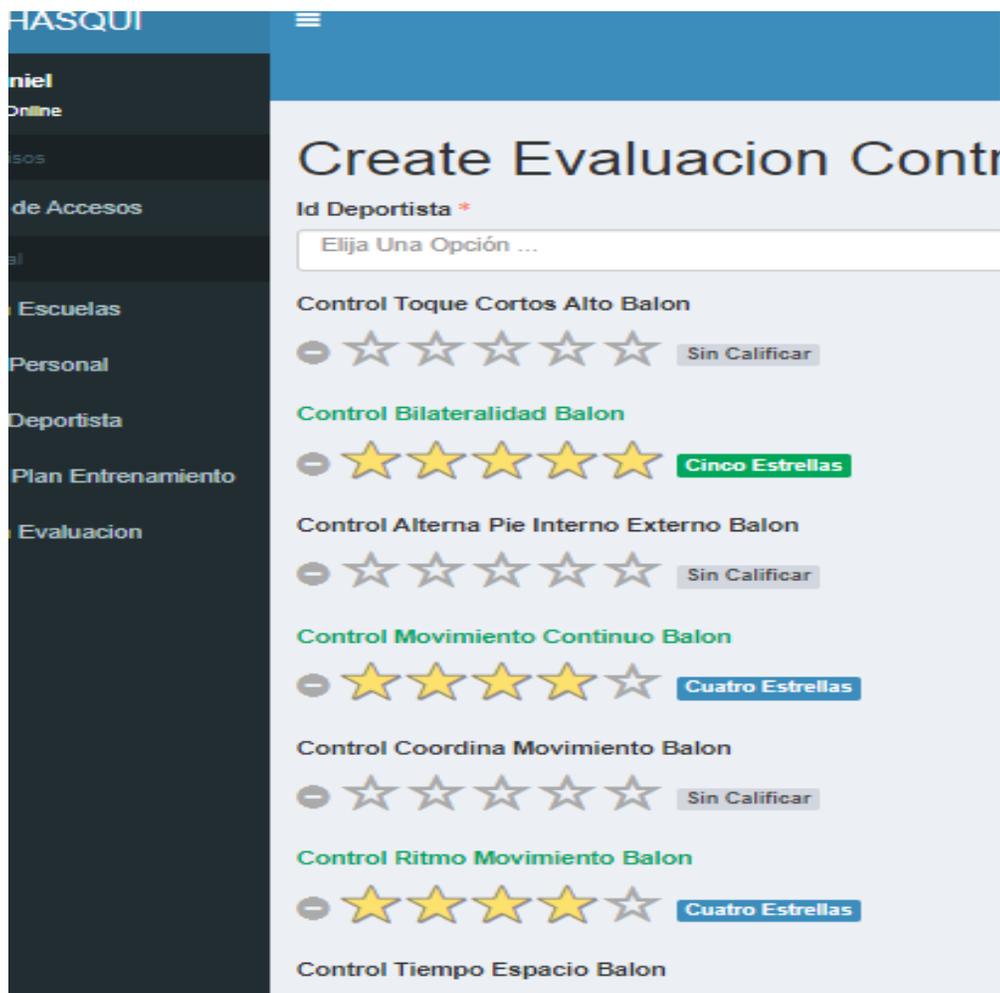
Mes	Mesociclo	Periodo
feb	1	Preparatorio - General
mar	2	Preparatorio - General

Sesion
Objetivo Sesion
Escuela
Personal
Plan
Microciclo
Fecha

Fuente: Elaboracion propia

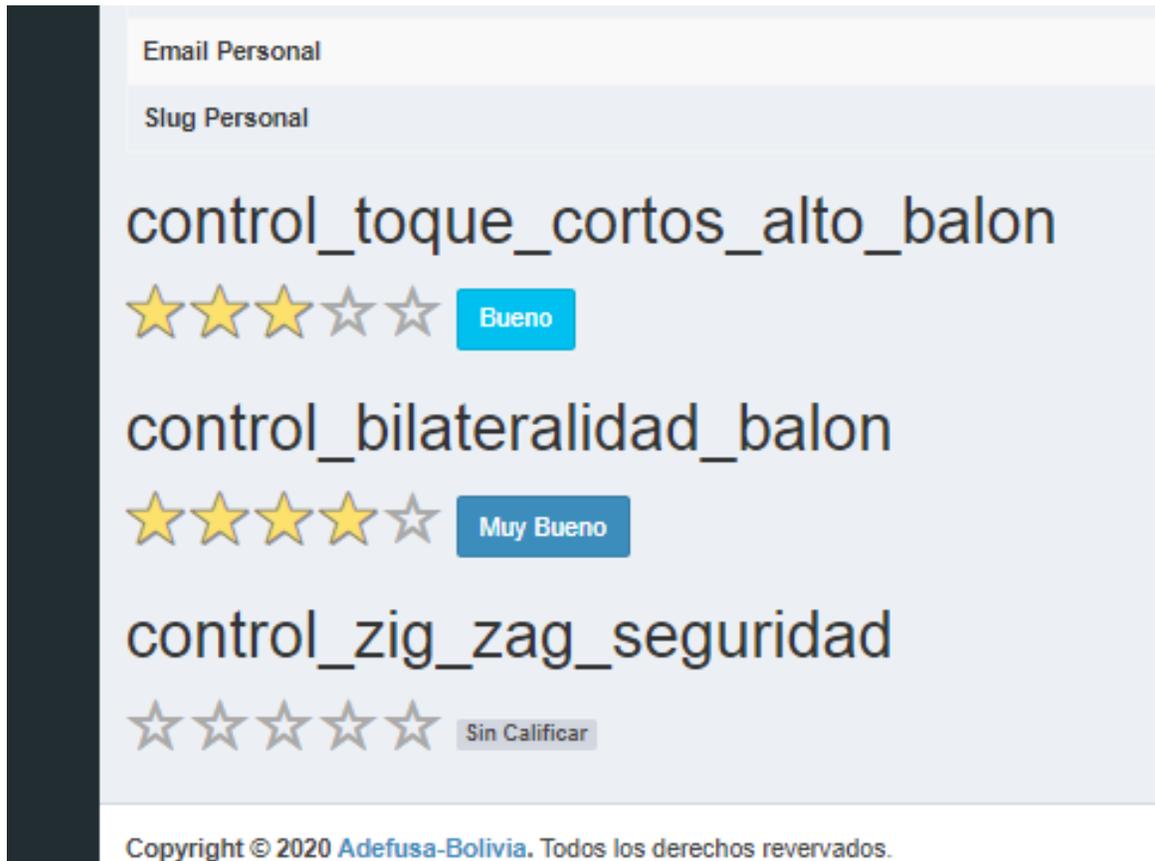
Módulo Evaluación Deportista. Este módulo muestra una valoración tipificada de uno a cinco estrellas, mostrando valores de cualificación desde malo hasta excelente.

Figura 3- 20 Evaluación Deportista



Fuente: Elaboracion propia

Figura 3- 21 Valoración Deportista



Fuente: Elaboracion propia

3.3.4. Pruebas.

Prueba de Caja Negra

Tabla 3- 4 Prueba Caja Negra Registrar Escuela

RF- 01	Registrar Escuela
--------	-------------------

DESCRIPCIÓN	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando alguien solicite REGISTRAR ESCUELA.	
ACTOR	Secretaría	
PRECONDICIÓN	El usuario debe ser admitido por el sistema, con el rol de secretaria.	
SECUENCIA NORMAL REGISTRAR ESCUELA	Paso	Acción
	1	El caso de uso comienza cuando la secretaria, indica “registrar”.
	2	El sistema muestra un formulario vacío para llenar datos de la nueva escuela.
	3	La secretaria (encargado), llena los datos requeridos por el formulario.
	4	El encargado indica guardar
	5	El sistema valida los datos ingresados por el encargado.
	6	El sistema guarda los datos.

	7	El sistema muestra un mensaje del proceso realizado.
	8	El caso de uso finaliza.
POS CONDICIÓN REGISTRAR ESCUELA	Se ha creado en el sistema, el Registro de Escuela Deportiva.	
FLUJO ALTERNATIVO REGISTRAR ESCUELA	Paso	Acción
	5	Si la validación de datos es errónea, muestra el mensaje de inválido. Vuelve a pedir el dato corregido.
Frecuencia esperada	2 veces/día	
Comentarios	La frecuencia será mayor al inicio de cada gestión al menos por 5 veces por día.	

Fuente: Elaboración Propia

Tabla 3- 5 Prueba Caja Negra Mantener Escuela

RF- 02	Mantener Escuela
---------------	------------------

DESCRIPCIÓN	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando alguien solicite MANTENER ESCUELA.	
ACTOR	Coordinador Escuelas Deportivas	
PRECONDICIÓN	El usuario debe ser admitido por el sistema, con el rol de Coordinador.	
AÑADIR ESCUELA		
SECUENCIA NORMAL AÑADIR ESCUELA	Paso	Acción
	1	El caso de uso comienza cuando el coordinador indica "Añadir".
	2	El sistema muestra un formulario vacío para llenar datos de la nueva escuela.
	3	El Coordinador, llena los datos requeridos por el formulario.
	4	El encargado indica guardar
	5	El sistema valida los datos ingresados por el encargado.

	6	El sistema guarda los datos.
	7	El sistema muestra un mensaje del proceso realizado.
	8	El caso de uso finaliza.
POS CONDICIÓN AÑADIR ESCUELA	Se ha añadido en el sistema, un Registro de Escuela Deportiva.	
FLUJO ALTERNATIVO AÑADIR ESCUELA	Paso	Acción
	5	Si la validación de datos es errónea, muestra el mensaje de inválido. Vuelve a pedir el dato corregir.
Frecuencia esperada	2 veces/semana	
Comentarios	La frecuencia será mayor al inicio de cada gestión al menos por 5 veces por semana.	
ACTUALIZAR ESCUELA		
SECUENCIA NORMAL ACTUALIZAR ESCUELA	Paso	Acción
	1	El caso de uso comienza cuando el Coordinador (usuario) indica "Actualizar".

	2	El sistema muestra una lista de Escuelas registradas.
	3	El sistema solicita el número de registro a actualizar o modificar.
	4	El usuario, ingresa el número de registro a actualizar o modificar.
	5	El usuario (Coordinador) modifica o actualiza los datos que desee.
	6	El usuario indica "Actualizar".
	7	El sistema valida los datos
	8	El sistema muestra un mensaje del proceso realizado.
	9	El caso de uso finaliza.
POS ACTUALIZAR ESCUELA	CONDICIÓN	Se ha actualizado en el sistema, un Registro de Escuela Deportiva.
	Paso	Acción

FLUJO ALTERNATIVO ACTUALIZAR ESCUELA	4	Si, el número de registro no existe, el sistema muestra mensaje "Registro no encontrado". Vuelve a pedir el registro correcto.
	7	Si la validación de datos es errónea, muestra el mensaje de inválido. Vuelve a pedir el dato para corregir.
Frecuencia esperada	3 veces/semana	
Comentarios	La frecuencia será mayor al inicio de cada gestión al menos por 5 veces por semana.	
ELIMINAR ESCUELA		
SECUENCIA NORMAL ELIMINAR ESCUELA	Paso	Acción
	1	El caso de uso comienza cuando el Coordinador (usuario) indica "Eliminar".
	2	El sistema muestra una lista de Escuelas registradas.
3	El sistema solicita el número de registro a eliminar.	

	4	El usuario, ingresa el número de registro a eliminar.
	5	El usuario indica "Eliminar".
	6	El sistema muestra un mensaje de confirmación
	7	El usuario confirma eliminación
	8	El sistema Elimina el registro.
	9	El caso de uso finaliza.
POS CONDICIÓN ELIMINAR ESCUELA	Se ha ELIMINADO en el sistema, un Registro de Escuela Deportiva.	
FLUJO ALTERNATIVO ELIMINAR ESCUELA	Paso	Acción
	4	Si, el número de registro no existe, el sistema muestra mensaje "Registro no encontrado". Vuelve a pedir el registro correcto.
	7	Si el usuario cancela la "eliminación", el sistema muestra el listado sin realizar acción alguna.

Frecuencia esperada	1 veces/semana
Comentarios	La frecuencia será mayor al inicio de cada gestión al menos por 3 veces por semana.

Fuente: Elaboración Propia

Tabla 3- 6 Prueba Caja Negra Mantener Personal

RF- 03	Mantener Personal Deportivo	
DESCRIPCIÓN	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando alguien solicite MANTENER PERSONAL.	
ACTOR	Coordinador Escuelas Deportivas	
PRECONDICIÓN	El usuario debe ser admitido por el sistema, con el rol de Coordinador.	
AÑADIR PERSONAL		
SECUENCIA NORMAL AÑADIR PERSONAL	Paso	Acción
	1	El caso de uso comienza cuando el coordinador indica "Añadir".

	2	El sistema muestra un formulario vacío para llenar datos del personal.
	3	El Coordinador, llena los datos requeridos por el formulario.
	4	El encargado indica guardar
	5	El sistema valida los datos ingresados por el encargado.
	6	El sistema guarda los datos.
	7	El sistema muestra un mensaje del proceso realizado.
	8	El caso de uso finaliza.
POS AÑADIR PERSONAL	CONDICIÓN	Se ha añadido en el sistema, un Registro del nuevo Personal.
FLUJO ALTERNATIVO AÑADIR PERSONAL	Paso	Acción
	5	Si la validación de datos es errónea, muestra el mensaje de inválido. Vuelve a pedir el dato a corregir.

Frecuencia esperada	2 veces/semana	
Comentarios	La frecuencia será mayor al inicio de cada gestión al menos por 2 por día.	
ACTUALIZAR PERSONAL		
SECUENCIA NORMAL ACTUALIZAR PERSONAL	Paso	Acción
	1	El caso de uso comienza cuando el Coordinador (usuario) indica "Actualizar".
	2	El sistema muestra una lista del Personal registrados.
	3	El sistema solicita el número de registro a actualizar.
	4	El usuario, ingresa el número de registro a actualizar.
	5	El usuario (Coordinador) actualiza los datos que desee.
	6	El usuario indica "Actualizar".
	7	El sistema valida los datos

	8	El sistema muestra un mensaje del proceso realizado.
	9	El caso de uso finaliza.
POS CONDICIÓN ACTUALIZAR PERSONAL	Se ha actualizado en el sistema, un Registro del Personal de la Escuela Deportiva.	
FLUJO ALTERNATIVO ACTUALIZAR PERSONAL	Paso	Acción
	4	Si, el número de registro no existe, el sistema muestra mensaje "Registro no encontrado". Vuelve a pedir el registro correcto.
	7	Si la validación de datos es errónea, muestra el mensaje de inválido. Vuelve a pedir el dato para corregir.
Frecuencia esperada	3 veces/semana	
Comentarios	La frecuencia será mayor al inicio de cada gestión al menos por 2 veces/día.	
ELIMINAR PERSONAL		
	Paso	Acción

SECUENCIA NORMAL ELIMINAR PERSONAL	1	El caso de uso comienza cuando el Coordinador (usuario) indica "Eliminar".
	2	El sistema muestra una lista de Escuelas registradas.
	3	El sistema solicita el número de registro a eliminar.
	4	El usuario, ingresa el número de registro a eliminar.
	5	El usuario indica "Eliminar".
	6	El sistema muestra un mensaje de confirmación
	7	El usuario confirma eliminación
	8	El sistema Elimina el registro.
	8	El caso de uso finaliza.
POS ELIMINAR PERSONAL	CONDICIÓN	Se ha eliminado en el sistema, el un registro del personal de la Escuela Deportiva.
	Paso	Acción

FLUJO ALTERNATIVO ELIMINAR PERSONAL	4	Si, el número de registro no existe, el sistema muestra mensaje "Registro no encontrado". Vuelve a pedir el registro correcto.
	7	Si el usuario cancela la "eliminación", el sistema muestra el listado sin realizar acción alguna.
Frecuencia esperada	1 veces/día	
Comentarios	La frecuencia será mayor al inicio de cada gestión al menos por 5 veces por día.	

Fuente: Elaboración Propia

Tabla 3- 7 Prueba de Caja Negra Designar Escuela

RF- 07	Designar Escuela Deportiva
DESCRIPCIÓN	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando alguien solicite "Designar Escuela".
ACTOR	Secretaría, Asistente.

PRECONDICIÓN	El usuario debe ser admitido por el sistema, con el rol de Asistente o Secretaría.	
AÑADIR DESIGNAR ESCUELA		
SECUENCIA NORMAL AÑADIR DESIGNAR ESCUELA	Paso	Acción
	1	El caso de uso comienza cuando La Secretaria o Asistente indica “Asignar Escuela”.
	2	El sistema muestra un listado de deportistas registrados.
	3	El sistema solicita un número de registro del deportista para asignar una escuela.
	4	El usuario, ingresa el número de registro del deportista.
	5	El sistema muestra datos del deportista
	6	El sistema añade una escuela al deportista.
	7	El sistema muestra un mensaje de proceso realizado.

	8	El caso de uso finaliza.
POS CONDICIÓN AÑADIR DESIGNAR ESCUELA	Se ha añadido en el sistema, un nuevo Registro de asignación Deportista a una escuela.	
FLUJO ALTERNATIVO AÑADIR DESIGNAR ESCUELA	Paso	Acción
	3	Si, el número de registro de deportista no existe, el sistema muestra mensaje “Registro no encontrado”. Vuelve a pedir el registro correcto.
	6	Si el número de registro de deportista ingresado está asignado una escuela, el sistema muestra un mensaje de “asignación de escuela realizada”. Vuelve a pedir otro registro de personal.
Frecuencia esperada	8 veces/día	
Comentarios	La frecuencia será mayor al inicio de cada gestión al menos por 15 veces por día.	
ACTUALIZAR DESIGNAR ESCUELA		
	Paso	Acción

<p>SECUENCIA NORMAL ACTUALIZAR DESIGNAR ESCUELA</p>	1	El caso de uso comienza cuando el La Secretaría o Asistente indica "Actualizar Asignación Deportista".
	2	El sistema muestra un listado de Deportistas Registradas.
	3	El sistema solicita el número de registro de Deportista a actualizar la designación.
	4	El usuario, ingresa el número de registro de deportista a actualizarla designación.
	5	El sistema muestra el registro a actualizar.
	6	El usuario actualiza la designación del deportista que desee.
	7	El usuario indica "Actualizar".
	8	El sistema muestra un mensaje del proceso realizado.
	9	El caso de uso finaliza.

POS ACTUALIZAR DESIGNAR ESCUELA	CONDICIÓN Se ha actualizado en el sistema, un Registro de designación del Deportista.	
FLUJO ALTERNATIVO ACTUALIZAR DESIGNAR ESCUELA	Paso	Acción
	4	Si, el número de registro de deportista no existe, el sistema muestra mensaje "Registro no encontrado". Vuelve a pedir el registro correcto.
Frecuencia esperada	5 veces/día	
Comentarios	La frecuencia será mayor al inicio de cada gestión al menos por 8 veces/día.	

Fuente: Elaboración Propia

Instalación

Requerimiento de Software y hardware. Para poder desarrollar el proyecto e implementarlo también necesitamos hardware y software, en la siguiente tabla se mencionan los requerimientos mínimos para el buen desempeño del sistema web planteado.

Tabla 3- 8 Cuadro de Requerimientos de Software y Hardware

FICHA DE INSTALACIÓN DE FRAMEWORK PHP
--

Rol responsable:	Administrador	
Nombre del framework:	Yii	
URL de descarga:	http://www.yiiframework.com/	
Versión:	2.0	
URL de instalación:	http://www.yiiframework.com/doc-2.0/guide-start-installation.html	
Requisitos de instalación:	Software:	PHP 5.4.0 o superior, apache última versión, un motor de base de datos
	Hardware:	Sistema básico / servidor web de contrato con licencias SSL y SSH.
Opciones o servicios que permite	<ul style="list-style-type: none"> - Implementa el modelo MVC (Model-View-Controller) - Yii tiene la filosofía de que el código debe ser escrito de una manera sencilla pero elegante. Yii nunca tratará de sobrecargar cosas principalmente con el propósito de seguir estrictamente cierto patrón de diseño - Proporciona generadores de consultas y ActiveRecord para ambas bases de datos relacionales y NoSQL - Soporte de almacenamiento en caché de varios niveles - Permite crear extensiones para distribuir - Proporciona alto rendimiento 	
Instrucciones de instalación:	<ol style="list-style-type: none"> 1. Crear una cuenta en https://github.com/ 2. Descargar composer. de https://getcomposer.org/Composer-Setup.exe e instalarlo 3. Los comandos 4 y 5 deben ejecutarse en una consola de símbolo de sistema y debe estar ubicado en el directorio webRoot 4. Instalar plugin de composer con el siguiente comando: <code>composer global require "fxp/composer-asset-plugin:1.0.0"</code> 5. Crear el primer sitio con yii a través de composer, ejecutando el siguiente comando. El proyecto se creará en la carpeta basic del directorio htdocs de xampp. El comando es el siguiente: <code>composer create-project --prefer-dist yiisoft/yii2-app-basic basic</code> 6. Después de la instalación con un navegador se puede verificar el proyecto Yii construido. La URL de prueba es la siguiente: http://localhost/basic/web/index.php 7. Si en pantalla del navegador aparece la aplicación, felicitaciones ya puedes trabajar con Yii 	

Dificultades en la instalación:	Confusión en el método de instalar el framework porque existen dos maneras, en esta ficha se dan las instrucciones para instalar Yii vía Composer
--	---

Fuente: Elaboración Propia

Para la implementación de una aplicación web existen diferentes tipos de hosting, pero por la complejidad al subir actualizaciones no son muy recomendables a menos que la Aplicación Web no vaya a tener actualizaciones futuras.

Las actividades que realizar están mediante la siguiente lista:

Tabla 3- 9 Actividades para la implementación

ACTIVIDAD	REALIZADO
Compra del dominio	SI
Compra de Web Hosting	SI
Configuración de Subdominio	SI
Instalación de las herramientas necesarias	PENDIENTE
Instalación de la Aplicación Web	PENDIENTE
Crear las migraciones de la base de datos	SI

Modificación de las variables de entorno en Yii2 para pasar a producción	PENDIENTE
Configurar Credenciales de Acceso remoto al Servidor	PENDIENTE

Fuente: Elaboración Propia

Una vez terminada las actividades pendientes la Aplicación Web estará disponible en línea las 24 horas.

3.3.5. Mantenimiento.

Con el fin de corregir errores en producción y subsanar problemas o realizar cambios a fin de conservar la capacidad de prestar el servicio para el cual fue diseñado. Se realizará el mantenimiento en los siguientes aspectos:

Preventivo. Modificando el software de manera proactiva a fin de corregir algunos códigos sueltos, para mejorar la comprensión y optimizar el funcionamiento y el consumo de recursos.

Correctivo. Actuando activamente frente errores que aparezcan durante el uso cotidiano del sistema.

Calidad y Seguridad

4.1. Calidad

El objetivo es desarrollar un software de calidad, asegurándose que cumpla con todas las expectativas del usuario, para quien se desarrolla el sistema. Y si este cumple entonces será denominado como un software de calidad.

4.2. Metodología de evaluación calidad

De acuerdo con la propuesta, aplicaremos la técnica **ISO 9126**, la calidad de producto queda definida a un alto nivel de abstracción por las características denominadas *usabilidad, funcionalidad, confiabilidad, eficiencia, mantenibilidad y portabilidad*. Luego, se emplea un proceso de descomposición recursivo basado en sub características y atributos, en consideración de una meta de evaluación y perfil de usuario dados.

4.3. Fases

4.3.1. Usabilidad.

La usabilidad consiste en evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema ya sea en la fase de aprendizaje o ya en uso promedio.

En la siguiente tabla se observa estos criterios en niveles de porcentajes a los que llega el sistema en cuanto a su comprensibilidad, para el usuario, y posteriormente se da el porcentaje final de usabilidad del sistema.

Tabla 4- 1 Tabla de Pregunta de Usabilidad

	SI	NO

¿Aprendió a usar rápido el sistema?	9	1	90
¿Las pantallas que vio fueron de su agrado?	10	0	100
¿Las pantallas que vio fueron fáciles de comprender?	9	1	90
¿El sistema responde rápido a sus solicitudes?	9	1	90
¿El sistema facilita el trabajo?	9	1	90
¿Es fácil navegar por distintas opciones?	10	0	100
¿Las operaciones que realizan no son complicadas?	10	0	100
¿El sistema proporciono las respuestas requeridas?	9	1	100
¿El sistema reduce el tiempo en su trabajo?	10	1	100
¿El sistema no presento errores?	8	2	80
Promedio			94
Fuente: Elaboración Propia			

4.3.2. Funcionalidad.

La funcionalidad es la capacidad del software de proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas, este atributo del sistema no puede medirse forma directa, por esa razón para el cálculo de la funcionalidad utilizaremos la métrica de punto función, para esto se debe determinar cinco características de dominios de información. Los valores de información se definen de la siguiente forma.

- **Número de entradas de usuario.** Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada.
- **Número de salidas de usuario.** Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto la salida se refiere a informes, pantallas, mensajes de error, etc.
- **Número de peticiones de usuario.** Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.
- **Número de interfaces externas.** Se cuentan todas las interfaces legibles por la máquina (por ejemplo: archivos de datos) que se utilizan para transmitir información a otro sistema.

Para calcular los puntos función se usó la siguiente formula:

$$PF = Cuenta\ Total * (0.75 + 0.01 * \Sigma Fi)$$

Donde:

PF: Medida de funcionalidad.

Cuenta Total: Es la suma de los siguientes datos: N° de Entradas, N° de Salidas, N° de Peticiones, N° de Archivos y N° de Interfaces externas.

0.75: Confiabilidad del proyecto, varía de 1% al 100% (0 a 1).

0.01: Error mínimo aceptable de complejidad.

ΣFi : Son los valores de ajuste de complejidad, donde ($1 \leq i \leq 14$).

Analizando todas las interfaces que tiene el sistema se obtuvieron los siguientes datos:

Tabla 4- 2 Tabla de funcionalidad

PARÁMETRO DE MEDIDA	CANTIDAD
Nº de entradas de usuario	14
Nº de salidas de usuario	14
Nº peticiones de usuario	18
Nº de interfaces externas	3

Fuente: Elaboración Propia

Una vez obtenida la información en la tabla anterior, se procedió a calcular la cuenta total con el factor de ponderación media, que muestra la siguiente tabla:

Tabla 4- 3 Factor de Ponderación

PARÁMETRO DE MEDIDA	CANTIDAD		FACTOR DE PONDERACIÓN	TOTAL
Nro. De entradas de usuario	14	*	4	56
Nro. De salidas de usuario	14	*	5	70

Nro. De peticiones de usuario	18	*	7	126
Nro. De interfaces externas	3	*	10	30
TOTAL				282

Fuente: Elaboración Propia

La cuenta total de los puntos de función obtenidos se debe ajustar en función a las características ambientales del sistema. Los valores de ajuste de complejidad Fi basados en las respuestas a las preguntas formuladas de la siguiente tabla:

Tabla 4- 4 Puntos de Función

Nro.	Factores	Sin influencia 0	Incidental 1	Moderado 2	Medio 3	Significativo 4	Esencial 5	Fi
1	¿Requiere el sistema copias de seguridad y de recuperación fiables?						X	5
2	¿Se requiere comunicación de datos?						X	5
3	¿Existen funciones de procesos distribuidos?			X				2
4	¿Es crítico el rendimiento?		X					1
5	¿Será ejecutado el sistema en un SO existente y fuertemente utilizado?			X				2

6	¿Requiere el sistema entrada de datos interactiva?					X		4
7	¿Requiere la entrada de datos interactiva que se utilicen varias pantallas o varias operaciones?			X				2
8	¿Se utilizan los archivos maestros de forma interactiva?				X			3
9	¿Son complejas las entradas, las salidas y/o las peticiones?					X		4
10	¿Es complejo el procesamiento interno?						X	5
11	¿Se ha diseñado el código para ser reutilizable?					X		4
12	¿Están incluidas en el diseño la conversión y la instalación?				X			3
13	¿Se ha diseñado el sistema para soportar diferentes instalaciones en						X	5

	diferentes organizaciones?							
14	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?						x	5
Factor de ajuste de complejidad								50

Fuente: Elaboración Propia

$$PF = Cuenta\ Total * (0.65 + 0.01 * \Sigma Fi)$$

$$PF = 282 * (0.65 + 0.01 * 50)$$

$$PF = 282 * (1.15)$$

$$PF = 324.3$$

Para comparar los puntos función con su valor máximo, se calculó los puntos función con los valores de ajuste de complejidad al máximo que es en total el valor 70:

$$PF = Cuenta\ Total * (0.65 + 0.01 * \Sigma Fi)$$

$$PF = 282 * (0.65 + 0.01 * 70)$$

$$PF = 282 * (1.35)$$

$$PF = 380.7$$

Después de haber calculado ambos valores se tiene que la funcionalidad real es:

$$Funcionalidad = (324.3/380.7) * 100 \%$$

$$Funcionalidad = 85.2 \%$$

Ese resultado atribuye a que el Sistema de Gestión y Seguimiento Técnico para Escuelas Deportivas de El Alto: Escuelas Deportivas de Fútbol Fundación ADEFUSA, satisfecha a las necesidades explícitas en un 85.2% y un 14.8% no satisfecha a dichas necesidades, es decir que: La Aplicación Web cumple con los requisitos mínimos funcionales de forma satisfactoria.

4.3.3. Confiabilidad.

La confiabilidad es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizando en condiciones específicas, por cierto, tiempo. Para este punto se hizo el análisis del nivel de confiabilidad del sistema, para lo cual primeramente se considera la confiabilidad de cada módulo o subsistema de forma independiente.

Para calcular la confiabilidad de cada módulo se usó de la formula $R(t) = e^{-\lambda t}$.

Donde:

$R(t)$: Confiabilidad de un componente o subsistema t.

λ : Tasa de constantes de fallo ($\lambda = \text{N}^\circ \text{ de fallas de acceso} / \text{N}^\circ \text{ total de accesos al sistema}$).

t: Periodo de operación de tiempo.

$e^{-\lambda t}$: Probabilidad de falla de un componente o subsistema en el tiempo t.

Luego de realizar pruebas de cada módulo en un tiempo de 4 horas continuas se logró llenar la siguiente tabla:

Tabla 4- 5 Módulos a evaluar Confiabilidad

N°	Módulo	λ	t	R(t)
1	Escuelas	0.013	2 horas	0.97
2	Personal	0.028	2 horas	0.95
3	Plan Entrenamiento	0.025	2 horas	0.95
4	Deportista	0.015	2 horas	0.97
5	Control de Accesos	0.031	2 horas	0.97

Fuente: Elaboración Propia

Para calcular la confiabilidad del sistema completo, se deduce que si falla la autenticación (Control del Accesos) falla no se puede acceder a los demás módulos,

por lo tanto será conectado en serie con los demás módulos y el resto de los módulos serán conectado en paralelo ya que esto funcionan independientemente del otro. Por ello la confiabilidad del sistema estaría dada por la siguiente fórmula:

$$Conf = RS * Rp.$$

$$\text{Donde } R_s = R_1 = \sum_{i=2}^5 (R_i * P_i) / \sum_{i=2}^5 P_i$$

En la fórmula de R_p , la variable P_i es la participación del equipo en el desarrollo del módulo y como la participación fue al 100% = 1 se tiene el siguiente resultado:

$$R_p = (0.97+0.97-0.95+0.95)/4 = 3.84/4 = 0.96$$

Por lo tanto, la confiabilidad del sistema está dada por:

$$Conf = 0.97 * 0.96 = 0.93 = 93\%$$

De lo cual se puede decir que existe un 7% de probabilidad de que el sistema presente algún fallo cuando se exceda un tiempo de uso continuo, debido a que pueden existir fallas con la conexión del sistema a la base de datos, conexión del cliente al sistema, uso incorrecto del sistema por parte del usuario, errores en la entrada de datos.

4.3.4. Mantenibilidad

La Mantenibilidad es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales, para poder medir la calidad del mantenimiento del sistema utilizaremos el índice de madurez del software (IMS), que indica la estabilidad de un producto de software. El índice de madurez del software se calcula con la siguiente formula:

$$IMS = [M_t - (F_a + F_b + F_c)] / M_t$$

Donde:

M_t : Número de módulos en la versión actual

F_a : Número de módulos en la versión actual que se han cambiado

Fb: Número de módulos en la versión actual que se han añadido

Fc: Número de módulos en la versión anterior que se han borrado en la versión actual.

Recopilando la información requerida por la formula se obtuvo la información que se muestra en la siguiente tabla:

Tabla 4- 6 Factores de Madurez

Información	Valor
<i>Mt</i>	5
<i>Fa</i>	0
<i>Fb</i>	0
<i>Fc</i>	0

Fuente: Elaboración Propia

Ahora calculamos el IMS, usando los valores obtenidos:

$$IMS = [5 - (0 + 0 + 0)]/5$$
$$IMS = 5/5 = 100\%$$

Con ese resultado se concluye que el Sistema de Gestión y Seguimiento Técnico para Escuelas Deportivas de El Alto: Escuelas Deportivas de Fútbol Fundación ADEFUSA, tiene un índice de madurez de software del 100%.

4.3.5. Portabilidad.

La portabilidad es la capacidad que tiene el software para ser trasladado de un entorno a otro. Para poder medir la portabilidad del sistema usaremos la siguiente formula que indica el grado de portabilidad que tiene un software.

$$GP = 1 - (ET/ER)$$

Donde:

ET: Es la medida de los recursos necesarios para llevar el sistema a otro entorno.

ER: Es la medida de los recursos necesarios para crear el sistema en el entorno residente.

Si $GP > 0$, la portabilidad es más rentable que el re-desarrollo

Si $GP = 1$, la portabilidad es perfecta

Si $GP < 0$, el re-desarrollo es más rentable que la portabilidad.

Para llevar el sistema web a otro entorno se necesita una memoria extraíble de 8gb o más capacidad, para crear el sistema en el entorno residente se necesita inicialmente al menos 2 servidores dedicados o compartidos con acceso SSH, con sistema operativo este puede ser cualquiera (Windows, Linux o Mac OS en sus diferentes versiones), y servidor Apache, el lenguaje de programación PHP, el gestor de base de datos MySQL los cuales deben estar instalados en los servidores. Con esta información requerida por la fórmula, se procede a calcular el grado de portabilidad:

$$GP = 1 - (1/8) = 1 - 0.13 = 87,5\%$$

Por lo que se concluye que el sistema web tiene un grado de portabilidad del 87,5%.

4.4. Seguridad de Software

La seguridad lógica es muy importante durante el desarrollo del software, implementando la autenticación de usuarios y la visualización de las acciones permitidas a dicho usuario, que pertenecen al módulo de acceso a usuarios del sistema. Para proteger los datos se traslada dicha información de manera encriptado, también se aplica las validaciones para los datos de entrada que tiene el sistema y finalmente toda la seguridad que ofrece el framework Yii2.

4.4.1 Autenticación.

El acceso al sistema es controlado por la autenticación, en el cual un usuario debe introducir sus credenciales correctas (usuario y contraseña), estos datos son validados mediante código JavaScript, del lado del cliente quien controla la sintaxis de los campos y también de lado del servidor mediante el código PHP, que es la instancia que captura y valida los datos introducidos

Figura 4- 1 Validación

```
public function loginAdmin()
{
    if (($this->validate()) && $this->getUser()->role_id=ValueHelpers::getRoleValue('UserAdmin') &&
        $this->getUser()->status_id=ValueHelpers::getStatusValue('Active'))
    {
        return Yii::$app->user->login($this->getUser(),
            $this->rememberMe ? 3600 * 24 * 30 : 0);
    } else {

        throw new NotFoundHttpException('Acceso denegado.');
```

Fuente: Elaboracion propia

4.4.2. Seguimiento a las Acciones de los Usuarios

Una medida de seguridad es seguir algunas acciones que realizan los usuarios dentro del sistema, en este proyecto se aplica el seguimiento de acciones que ofrece el framework Yii2, mostrando todas las acciones que se tiene dentro del DEBUG, del sistema.

Figura 4- 2 Seguimiento Acciones del usuario

Mostrando 1-50 de 57 elementos.

#	Tag	Time	Processing Time	Peak Memory	Ip	Query Count	Mail Count	Method	Ajax	URL
1	5f932ca1b5e00	2020-10-23 21:18:57	160 ms	7.540 MB	:::1	38	0	GET	No	http://localhost/yii2build/backend/web/index.php
2	5f932c9ba83c6	2020-10-23 21:18:51	376 ms	7.633 MB	:::1	3	0	GET	No	http://localhost/yii2build/backend/web/index.php/gii
3	5f923fb3d3d45	2020-10-23 04:28:03	126 ms	9.575 MB	:::1	44	0	GET	No	http://localhost/yii2build/backend/web/index.php/personal/create
4	5f923f6dd2874	2020-10-23 04:26:53	145 ms	9.575 MB	:::1	44	0	GET	No	http://localhost/yii2build/backend/web/index.php/personal/create
5	5f923f4c76198	2020-10-23 04:26:20	125 ms	9.575 MB	:::1	44	0	GET	No	http://localhost/yii2build/backend/web/index.php/personal/create
6	5f923f1fe47c3	2020-10-23 04:25:35	88 ms	7.621 MB	:::1	6	0	GET	No	http://localhost/yii2build/backend/web/index.php/personal/create
7	5f923ecec60081	2020-10-23 04:24:14	131 ms	9.575 MB	:::1	44	0	GET	No	http://localhost/yii2build/backend/web/index.php/personal/create
8	5f923eba4ca91	2020-10-23 04:23:54	130 ms	9.575 MB	:::1	44	0	GET	No	http://localhost/yii2build/backend/web/index.php/personal/create
9	5f923e86d42cf	2020-10-23 04:23:02	88 ms	7.644 MB	:::1	6	0	GET	No	http://localhost/yii2build/backend/web/index.php/personal/create
10	5f923e4339d0b	2020-10-23 04:21:55	174 ms	9.575 MB	:::1	44	0	GET	No	http://localhost/yii2build/backend/web/index.php/personal/create
11	5f923e4154e4c	2020-10-23 04:21:53	337 ms	11.804 MB	:::1	53	0	GET	No	http://localhost/yii2build/backend/web/index.php/personal/index
12	5f923e2e4daf	2020-10-23 04:21:32	2.300 ms	10.145 MB	:::1	42	0	GET	No	http://localhost/yii2build/backend/web/index.php/plan-entrenamiento

Fuente: Elaboracion propia

4.4.3. Encriptación

La encriptacion de datos es muy importante para proteger los lotes de datos que se transmiten en los servidores. Para ello se uso algoritmos de encriptacion tales como MD5, SHA1 y AES, en el siguiente cuadro muestra com el framework llama a una funcion que realiza tal funcion.

Figura 4- 3 Encriptación

```
public function validatePassword($password)
{
    return Yii::$app->security->validatePassword($password, $this->password_hash);
}

/**
 * Generates password hash from password and sets it to the model
 */
@param string $password
*/
public function setPassword($password)
{
    $this->password_hash = Yii::$app->security->generatePasswordHash($password);
}

/**
 * Generates "remember me" authentication key
 */
public function generateAuthKey()
{
    $this->auth_key = Yii::$app->security->generateRandomString();
}

/**
 * Generates new password reset token
 */
public function generatePasswordResetToken()
{
    $this->password_reset_token = Yii::$app->security->generateRandomString() . '_' . time();
}

public function generateEmailVerificationToken()
{
    $this->verification_token = Yii::$app->security->generateRandomString() . '_' . time();
}
```

Fuente: Elaboracion propia

4.4.4. Seguridad en la base de datos

El gestor de base de datos Mysql, tiene sus propias medidas de seguridad como una básica autenticación, el manejo de roles, etc. Uno de los ataques más frecuentes realizados a una base de datos son las inyecciones sql. Las inyecciones SQL son cadenas de instrucciones SQL que un usuario puede introducir en cualquier campo de entrada de un formulario. El framework Yii2 está equipado para evitar este tipo de ataques ya que para hacer consultas este nos

pide hacer las consultas por medio de funciones que pueden ser usadas solamente por los modelos, estas funciones internas del framework proveen protección contra las inyecciones SQL y están destinadas a realizar solo un tipo de acción. En la siguiente tabla se muestra la explicación de las funciones:

Tabla 4- 7 Funciones Mysql Yii2

FUNCIÓN	DESCRIPCIÓN
Execute()	Para ejecutar un SQL no consulta (como INSERT, DELETE, UPDATE)
queryAll(), Model::find();	Ejecuta la sentencia SQL y devuelve todas las filas a la vez.
queryOne()	Ejecuta la sentencia SQL y devuelve la una fila del resultado.

Fuente: www.yiiframework.com

Análisis de Costos

4.5. Modelo Cocomo II

El Modelo de Construcción de Costo COCOMO (Constructive Cost MOdel), es un modelo que se utiliza para la estimación de costos de un software. Incluye tres sub-modelos, cada uno ofrece un nivel de detalle y aproximación. En los tres modelos de estimación se utilizan las tres siguientes ecuaciones:

$$E = a(KLDC)^b * m(X), \text{ en personas mes}$$

$$D = c(E)^d, \text{ en meses}$$

$$P = \frac{E}{D}, \text{ en personas}$$

Donde:

E: Esfuerzo requerido por el proyecto, en meses.

D: Tiempo requerido por el proyecto, en meses.

P: Número de personas requeridos por el proyecto.

a, b, c y d: Constantes con valores definidos, según cada sub-modelo.

KLDC: Cantidad de líneas de código, en miles.

m(X): Multiplicador que depende de 15 atributos

A la vez cada modelo se divide en modos, estos modos son:

- **Modo orgánico:** es un pequeño grupo de programadores experimentados desarrollando software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas docenas de miles (medio).
- **Modo semi – acoplado:** Corresponde a un esquema intermedio entre el orgánico y el rígido, el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.

- **Modo rígido o empotrado:** El proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único, siendo difícil basarse en la experiencia puesto que puede no haberla.

Tabla 5- 1 Coeficiente del Modelo COCOMO II

PROYECTO DE SOFTWARE	a_b	b_b	c_b	d_b
Orgánico	2.4	1.05	2.2	0.38
Semi – acoplado	3.0	1.12	2.5	0.35
Empotrado	3.80	1.20	2.5	0.32

Fuente: (Pressman, 2010)

4.6. Cálculo de Costo

Para el cálculo del desarrollo del software se tendrá como partida el punto función no ajustado valor ya encontrado en el capítulo anterior. Recordando es el siguiente:

$$PF = 324.3$$

Este resultado se debe convertir a KLDC (Kilos de Líneas de Código), para ello se utiliza la siguiente la tabla.

Tabla 5- 2 Conversión de puntos de Función a KDLC

LENGUAJE	NIVEL	FACTOR LDC/PF
C	2.5	128

ANSI BASIC	5	64
JAVA	6	53
PL/I	4	80
ANSI COBOL 74	3	107
VISUAL BASIC	7.00	46
ASP	9.00	36
PHP	11.00	29
VISUAL C++	9.50	34

Fuente: (Pressman, 2010)

Calculando las líneas de código en la siguiente ecuación tenemos:

$$LDC = PF * \frac{\text{Factor LDC}}{PF}$$

$$LDC = 324.3 * 29$$

$$\mathbf{LDC = 9404.7}$$

Para convertirlo a KLDC dividimos LDC entre 1000. Calculando el número de líneas distribuidas en el sistema KLCD se tiene:

$$KLDC = \frac{\text{total LDC}}{1000}$$

$$KLDC = \frac{9404.7}{1000}$$

$$\text{KLDC} = 9.5$$

A continuación, haremos el cálculo del esfuerzo necesario para la programación del sistema. La ecuación que nos ayudara a hallar el esfuerzo, viene dada de la siguiente manera:

$$E = a * (\text{KLDC})^b, \text{ en personas/mes}$$

Donde:

E: Es el esfuerzo expresado en personas por mes.

a y b: Son constantes.

KLDC: Es un número estimado de código fuente en miles distribuidas.

Como nuestro proyecto es del tipo Orgánico, utilizaremos $a = 2.4$ y $b = 1.05$. Reemplazando estos valores en la ecuación, tenemos:

$$E = 2.4 * (9.5)^{1.05}$$

$$E = 25,6 \left(\frac{\text{personas}}{\text{mes}} \right)$$

Calculando D con $c = 2.2$ y $d = 0.38$ tenemos:

$$D = c(E)^d \text{ en meses}$$

$$D = 2.2 * (25.6)^{0.38}$$

$$D = 7,55 \cong 8 \text{ (meses)}$$

El proyecto deberá tener un desarrollo de aproximadamente de 8 meses.

Aplicando en la ecuación para el cálculo del personal requerido en este caso el Número de programadores para el desarrollo es:

$$P = \frac{E}{D}, \text{ en personas}$$

$$P = \frac{25.6}{7.55}$$

$$P = 3.40 \cong 4(\text{programadores})$$

El salario promedio de un programador junior oscila entre los 2800 Bs y 3500 Bs., en nuestro caso tomaremos un promedio con un valor de 3150 Bs, a partir de este monto podemos calcular el costo del software desarrollado:

$$\text{Costo del software desarrollado} = P * \text{salario de un programador}$$

$$\text{Costo del software desarrollado por persona} = 4 * 3150$$

Costo del software desarrollado por persona = 12,600 Bs.

Como el desarrollo de software se lo estima en 8 meses tendremos:

$$\text{Costo total de desarrollo} = \text{Costo del software desarrollado} * \text{Numero de meses}$$

$$\text{Costo total de desarrollo} = 12600 * 8$$

Costo total de desarrollo = 100800 Bs.

Costo de Implementación del Software

Tabla 5- 3 Costo de Implementación

DETALLE	IMPORTE (Bs)
Dominio (12 meses)	50
VPS/SSD (12 meses)	550
TOTAL	600

Fuente: (Elaboración propia)

Costo de Elaboración del Software.

Los costos de elaboración del proyecto se refieren principalmente a los gastos que se realizan a lo largo de las diferentes fases de la metodología Web UWE, estos son los detallados a continuación:

Tabla 5- 4 Costo de Elaboración

DETALLE	IMPORTE (Bs)
Análisis y diseño del proyecto	1400
Material de escritorio	120
Internet	600
Otros	200
TOTAL	2,320

Fuente: (Elaboración propia)

Costo total del Software.

El costo total del software se lo obtiene de la sumatoria del costo de: desarrollo, implementación y elaboración del proyecto:

Tabla 5- 5 Costo Total

DETALLE	IMPORTE (Bs)
Costo de desarrollo	100,800
Costo de implementación	600
Costo de elaboración	2320
TOTAL	10,3720

Fuente: (Elaboración propia)

Por lo tanto, el costo total del proyecto es **10,3720Bs.** o su equivalencia en dólares americanos a una tasa de cambio de 6.96 es de 14,903 \$us.

Conclusiones y Recomendaciones

6.1. Conclusiones.

Una vez finalizado el proyecto de grado denominado “SISTEMA DE GESTIÓN Y SEGUIMIENTO TÉCNICO PARA ESCUELAS DEPORTIVAS DE EL ALTO”. Para el CASO: ESCUELAS DEPORTIVAS DE FUTBOL FUNDACIÓN ADEFUSA, se ha logrado alcanzar el objetivo principal planteado, cumpliendo con las necesidades mínimas de la Fundación.

Tomando en cuenta los objetivos planteados se llega a las siguientes conclusiones:

- Se desarrolló el Sistema de Gestión y Seguimiento Técnico para Escuelas Deportivas de El Alto, para la Escuela de Fútbol Adefusa en su totalidad, con los módulos requeridos por la Fundación.
- Se logró mejorar el tiempo empleado en la generación de planes de trabajo, sesiones de entrenamiento y entrega de reportes, ya que se realiza este proceso de forma más eficiente.
- Se logró que el usuario pueda acceder al sistema con un determinado privilegio por medio de una autenticación donde debe registrar con usuario y contraseña.
- Se logró mejorar el registro de deportistas por sedes, escuelas y categorías y se tiene las características detalladas de los mismos.
- Se logró disminuir los tiempos en la generación de reportes tanto de deportistas y plantel técnico, planes de entrenamiento de entrenadores por escuelas con sus diferentes actividades (sesiones), para así tener un mejor control los deportistas y sus planes de entrenamientos.

6.2. Recomendaciones.

A partir del presente trabajo se propone las siguientes recomendaciones, con el fin de buscar el mejoramiento del sistema:

- Se recomienda usuario Administrador, al dar de alta a un funcionario que debe registrarse con un email actual y valido, en las instancias de creación de nuevo personal y registro de personal como usuario, ésta debe ser obligatoriamente el mismo email.
- Se recomienda usuario Administrador que, al asignar un rol al personal, debe tener en cuenta que ésta debe ser activa y debe dar el rol y tipo de usuario respectivo al tipo de trabajo a realizar en el sistema.
- Se recomienda realizar copias de seguridad de la base de datos periódicamente, para llevar una copia de respaldo.
- Por razones de seguridad, se recomienda al usuario no compartir su contraseña, ya que en cada módulo que realizase cambios tendrá registrado su usuario y fecha de actualización.

Bibliografía

- BOMPA, T.O. (2000): Periodización del entrenamiento, T. O. (2000). *Periodización del entrenamiento deportivo*. Barcelona.: PAIDOTRIBO.
- Booch, G., Rumbaugh, J., & Jacobson, I. (1996). *El Lenguaje Unificado de Modelado*. Wesley.: Addison.
- Fleck, S. J., & Kraemer, W. J. (1996). *Periodization*. New York: Advanced Research Press, Inc.
- HARRE, D. (1987). *Teoría del entrenamiento deportivo*. Buenos Aires.: Stadium.
- LOPEZ LOPEZ, J. (2002). *160 fichas de entrenamiento para juveniles*. Sevilla.: WANCEULEN.
- Navarro, F. (1996). Hacia una conceptualización, F. (1996). Hacia una conceptualización del entrenamiento deportivo (Caso de deportes cíclicos). • *Navarro, F. (1996). Hacia una conceptualización del RED- Revista de Entrenamiento Deportivo.*, 17-24.
- Platonov, V. N. (1988). El entrenamiento deportivo, V. N. (1998). *El entrenamiento deportivo: Teoría y metodología*. Barcelona: Paidotribo.
- Pressman, R. S. (2010). *Ingeniería de Software un Enfoque Práctico*. Mexico: McGraw Hill.
- Rodríguez Gómez, G. A. (2003). Programa de Evaluación del Rendimiento Deportivo. *efDeportes*.
- Rodríguez Valencia, J. (2003). *Introducción a la Administración con Enfoque de Sistemas*. México: Editorial Thomson.

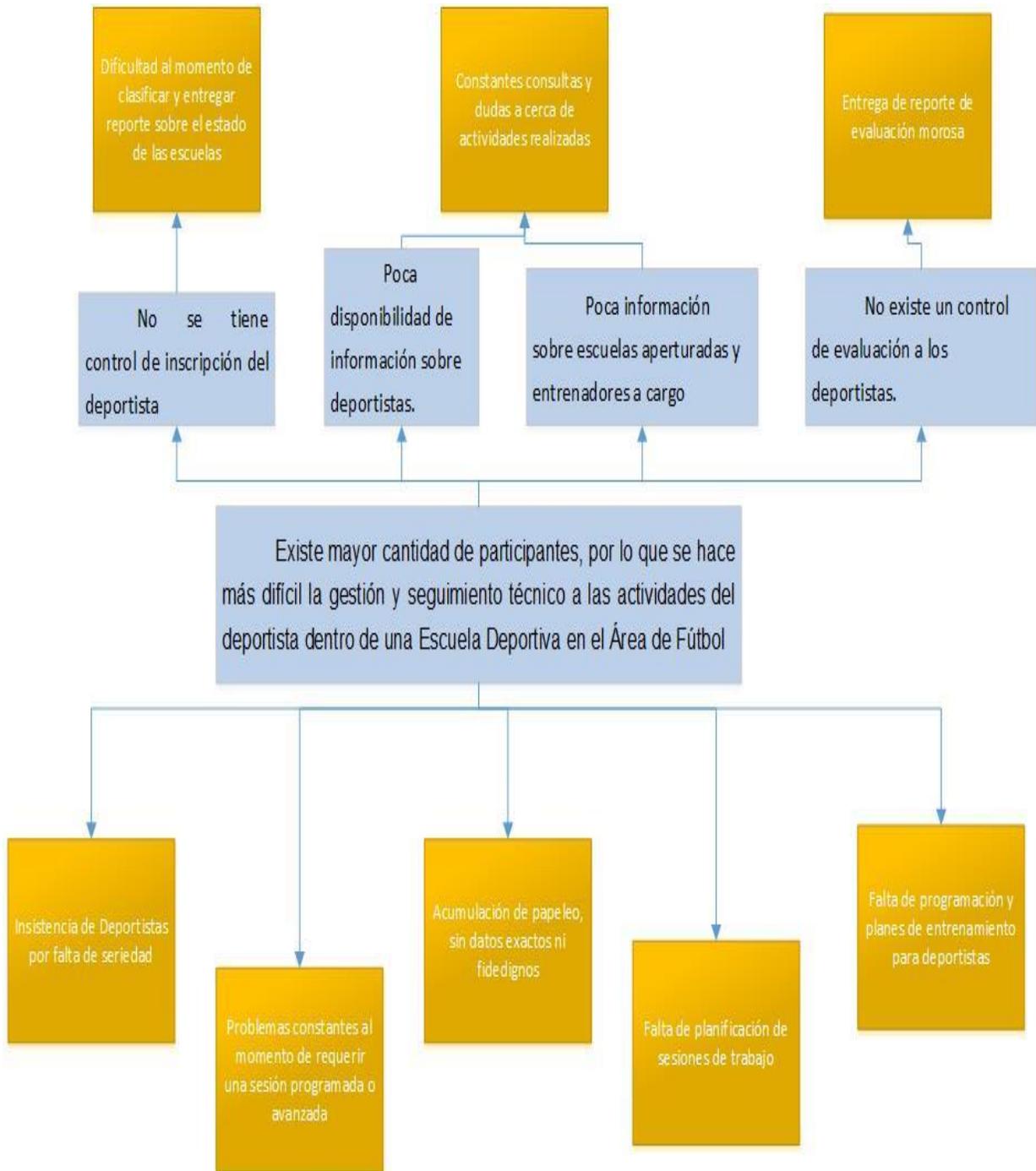
Rojas, C. (2013). *Adicción Deportiva*. Obtenido de <http://addeportiva-poli.blogspot.com/p/que-es.html>. Cristhian Rojas.

Schmuller, J. (1997). *Aprendiendo UML en 24 horas*. Mexico: Prentice Hall.

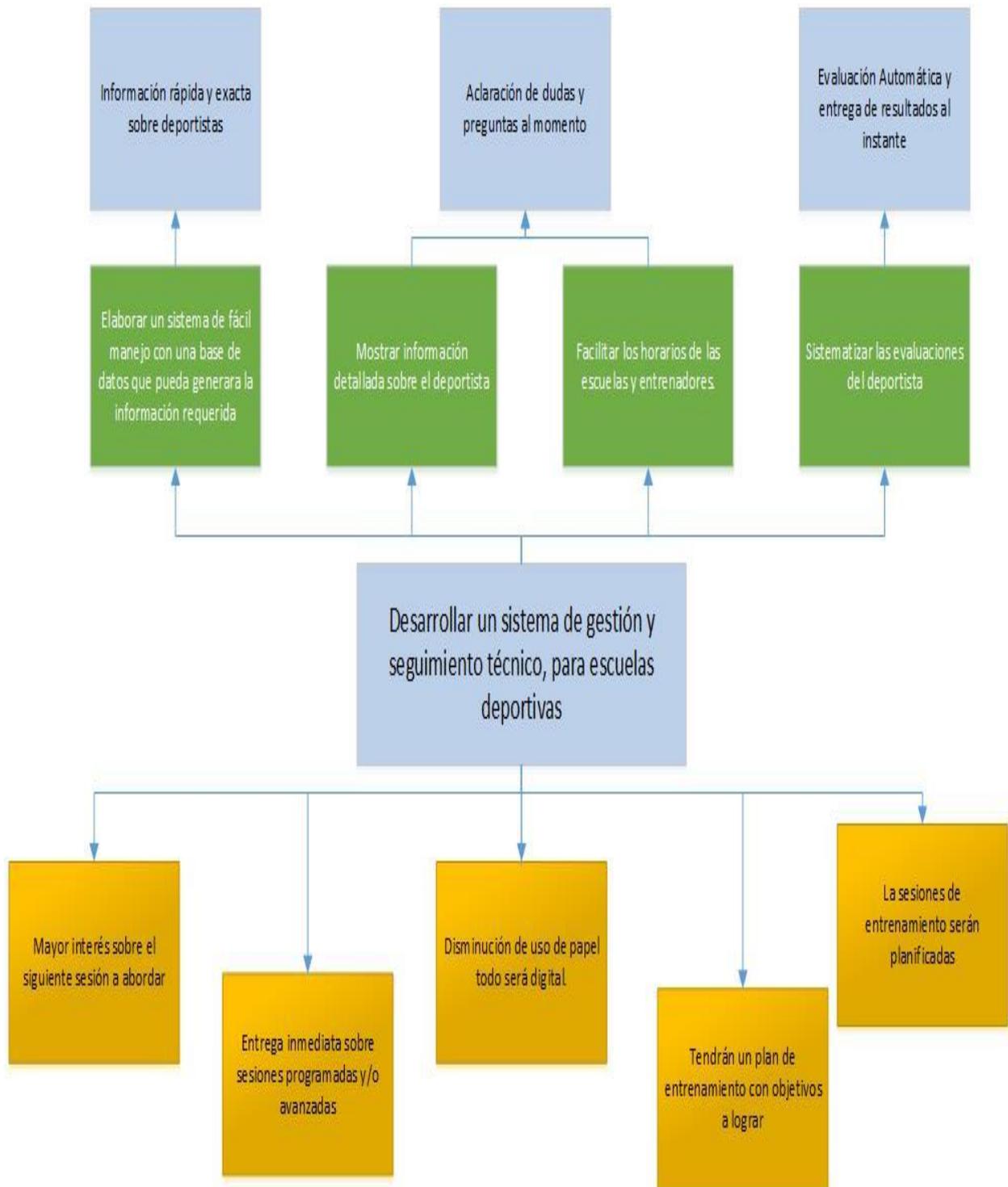
Verjoshanskij, I. V. (1990). *Entrenamiento deportivo. Planificación y programación*. Barcelona España: Martinez Roca.

Anexo A

A.1. Árbol de Problemas

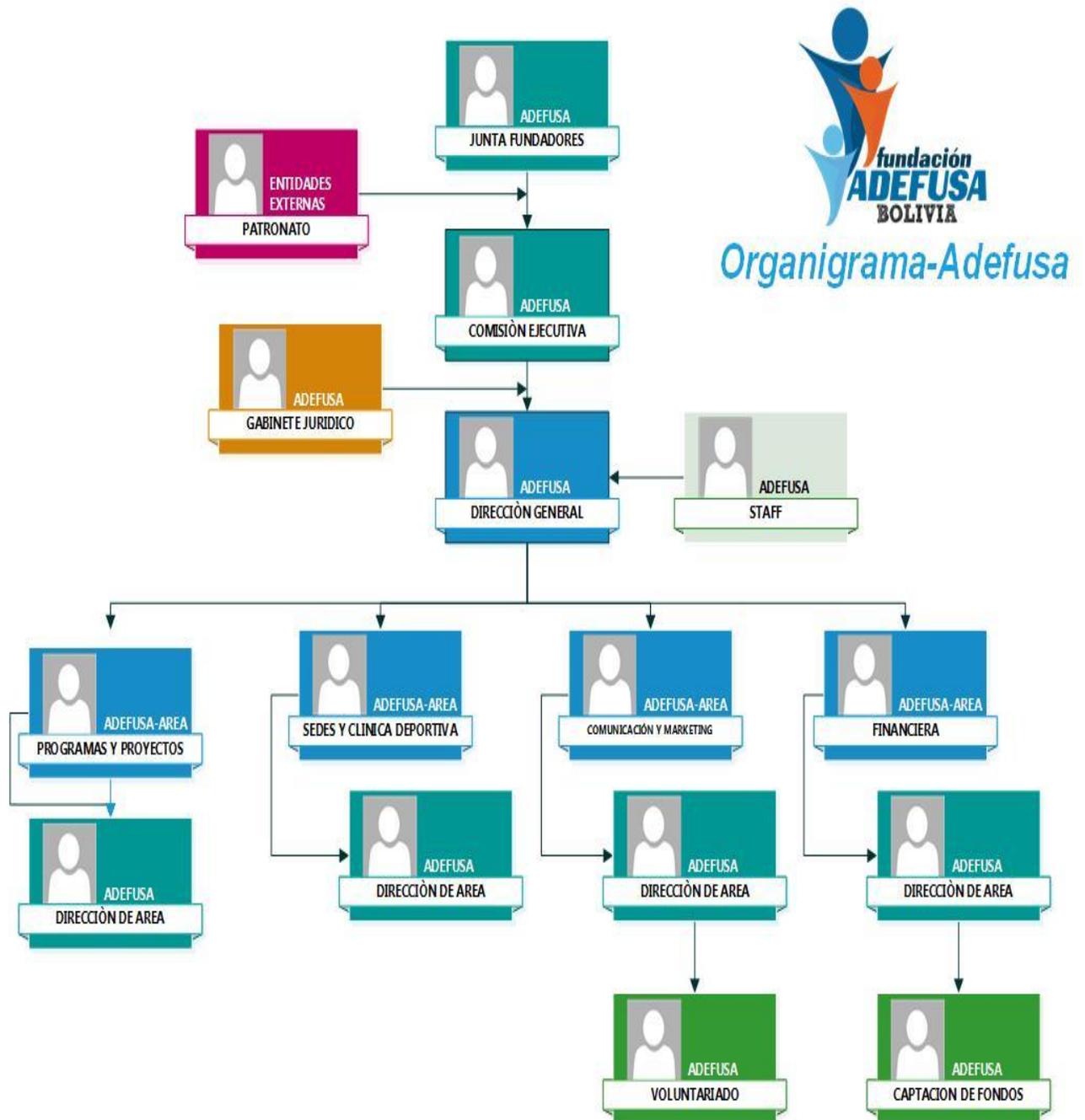


A.2. Árbol de Objetivos

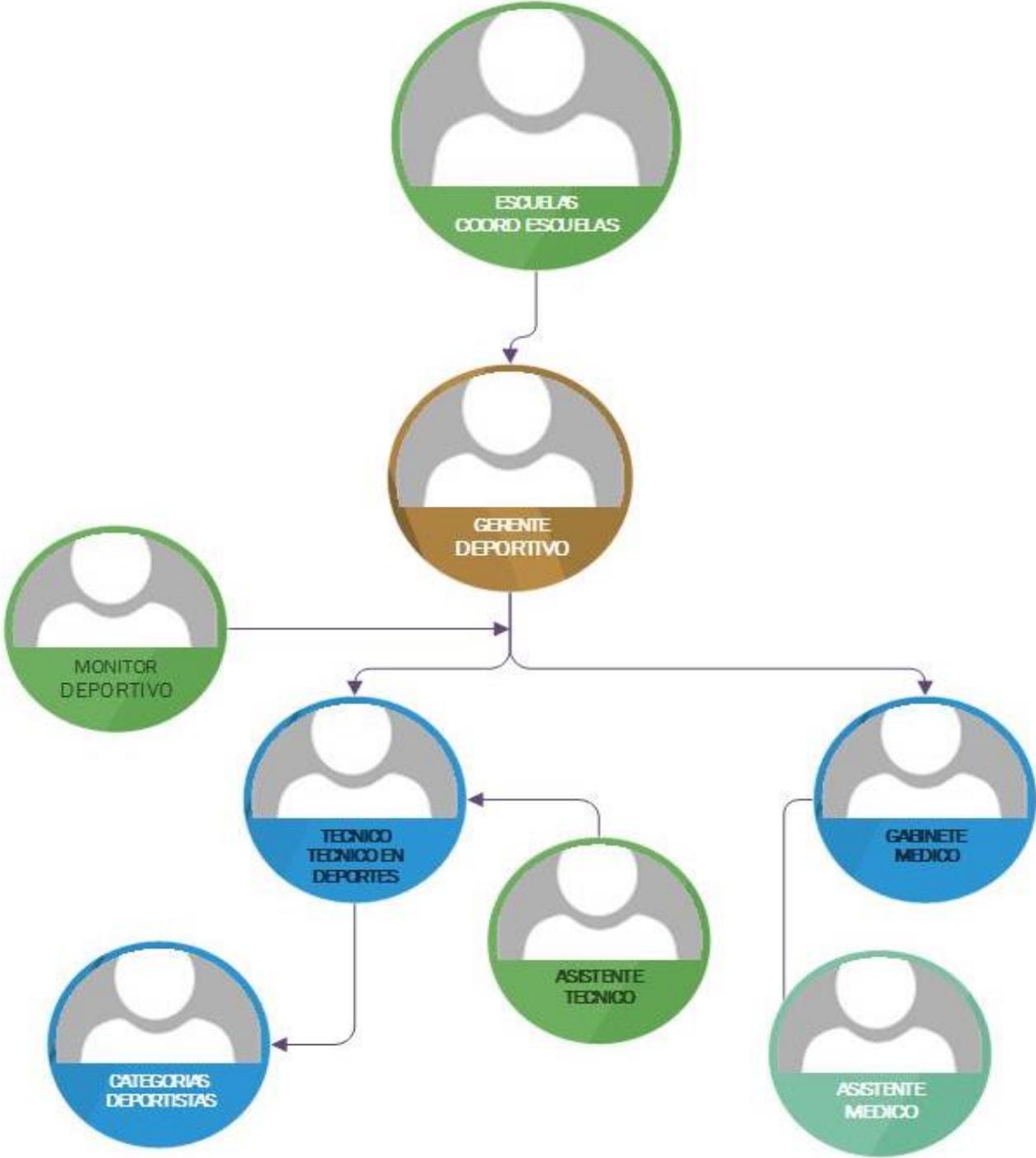


Anexo B

B.1. Organigrama Adefusa



B.2. Organigrama Escuelas Deportivas



B.3. Kárdex Deportista Adefusa

FEDERACION BOLIVIANA DE FUTBOL

Nº 094133

REGISTRO NACIONAL DE JUGADORES

Ficha Individual

Nombres del Jugador: Luis Miguel

Apellidos del Jugador: Condori Soliz
Ap. Paterno Ap. Materno

Integrará el Club: Adefusa

Categoría: Sub - 9

Asociación, Liga ó Liga Provincial: A.F.L.P.

Nacionalidad: Boliviana Est.Civil: Soltero

Lugar y fecha de Nacimiento: La Paz, 28 de junio de 2000

Nombre del padre: Marcial Condori Choque

Nombre de la madre: Vanessa Soliz Condori

Nombre de la esposa:

Edad: 8 años Carnet de Identidad: 706928040

Domicilio: Calle 11 No. 17 z. Anexo Villa Tunari

Procede del Club: NO REGISTRADO

Asociación o Liga:

REGISTRO F.B.F. 329695

Carnet del jugador Nº:

Lugar y Fecha: La Paz, 7 de mayo de 2008



Condori
FIRMA DEL JUGADOR

CERTIFICAMOS: Que los datos personales fueron extractados de los documentos que se indican y que el jugador dá su conformidad por lo que firma en presencia de los suscritos, de todo lo cuál se hace responsable al Club que representamos en la medida y con los alcances establecidos en el Reglamento del Estatuto Orgánico de la Federación Boliviana de Fútbol.

FIRMA *[Signature]*
NOMBRE Ruben Carlos Machaca
SECRETARIO DEL CLUB



FIRMA *[Signature]*
NOMBRE Tomas Machaca M.
PRESIDENTE DEL CLUB

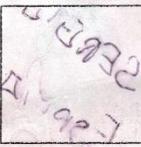
NOTA: A los efectos de control el juego completo debe ser enviado a la Federación Boliviana de Fútbol, quedándose con el original devolviendo las copias selladas; una para la Asociación o Liga y la otra para el Club.

B.4. Kárdex Deportista Adefusa Campeonato

Viceministerio de Deportes
Gobierno Municipal de El Alto
Academia de Fútbol Adefusa
El alto - Bolivia

II CAMPEONATO NACIONAL DE FÚTBOL INFANTIL SUB 13 - 2006 KARDEX Habilitación de Representativo Departamental

Departamento:	Esc. de Fútbol:	Tel.:
---------------	-----------------	-------

<p style="text-align: center;">II. CAMPEONATO NACIONAL DE FÚTBOL INFANTIL SUB 13 "EL ALTO - BOLIVIA 2006" Reg. N°</p>  <p>Nombres: <u>DANNY</u></p> <p>A. Paterno: <u>MARTINEZ</u></p> <p>A. Materno: <u>GONZALEZ</u></p> <p>Origen: <u>POTOSI</u></p> <p>F. Nacimiento: <u>22-05-95</u></p> <p>Edad: <u>10</u> Est:..... Peso:.....</p> <p style="text-align: center;"><i>Sello y Firma Comité Técnico</i></p>	<p style="text-align: center;">II. CAMPEONATO NACIONAL DE FÚTBOL INFANTIL SUB 13 "EL ALTO - BOLIVIA 2006" Reg. N°</p>  <p>Nombres: <u>IVAN</u></p> <p>B. Paterno: <u>ARMANDO</u></p> <p>A. Materno: <u>AGUILAR</u></p> <p>Origen: <u>POTOSI</u></p> <p>F. Nacimiento: <u>27-12-96</u></p> <p>Edad: <u>10</u> Est:..... Peso:.....</p> <p style="text-align: center;"><i>Sello y Firma Comité Técnico</i></p>
<p style="text-align: center;">II. CAMPEONATO NACIONAL DE FÚTBOL INFANTIL SUB 13 "EL ALTO - BOLIVIA 2006" Reg. N°</p>  <p>Nombres: <u>SERGIO</u></p> <p>C. Paterno: <u>ESPINOZA</u></p> <p>A. Materno: <u>TRABAJA</u></p> <p>Origen: <u>POTOSI</u></p> <p>F. Nacimiento: <u>23-09-94</u></p> <p>Edad: <u>12</u> Est:..... Peso:.....</p> <p style="text-align: center;"><i>Sello y Firma Comité Técnico</i></p>	<p style="text-align: center;">II. CAMPEONATO NACIONAL DE FÚTBOL INFANTIL SUB 13 "EL ALTO - BOLIVIA 2006" Reg. N°</p>  <p>Nombres: <u>CRISTIAN</u></p> <p>D. Paterno: <u>COLAZA</u></p> <p>A. Materno: <u>VILLAS</u></p> <p>Origen: <u>POTOSI</u></p> <p>F. Nacimiento: <u>13-03-94</u></p> <p>Edad: <u>12</u> Est:..... Peso:.....</p> <p style="text-align: center;"><i>Sello y Firma Comité Técnico</i></p>
<p style="text-align: center;">II. CAMPEONATO NACIONAL DE FÚTBOL INFANTIL SUB 13 "EL ALTO - BOLIVIA 2006" Reg. N°</p>  <p>Nombres: <u>RONE</u></p> <p>E. Paterno: <u>RAMIREZ</u></p> <p>A. Materno: <u>CRUZ</u></p> <p>Origen: <u>POTOSI</u></p> <p>F. Nacimiento: <u>22-02-94</u></p> <p>Edad: <u>12</u> Est:..... Peso:.....</p> <p style="text-align: center;"><i>Sello y Firma Comité Técnico</i></p>	<p style="text-align: center;">II. CAMPEONATO NACIONAL DE FÚTBOL INFANTIL SUB 13 "EL ALTO - BOLIVIA 2006" Reg. N°</p>  <p>Nombres: <u>FLAMINIO</u></p> <p>F. Paterno: <u>MAMANI</u></p> <p>A. Materno: <u>MAMANI</u></p> <p>Origen: <u>POTOSI</u></p> <p>F. Nacimiento: <u>23-03-94</u></p> <p>Edad: <u>12</u> Est:..... Peso:.....</p> <p style="text-align: center;"><i>Sello y Firma Comité Técnico</i></p>
<p style="text-align: center;">II. CAMPEONATO NACIONAL DE FÚTBOL INFANTIL SUB 13 "EL ALTO - BOLIVIA 2006" Reg. N°</p>  <p>Nombres: <u>JESUS</u></p> <p>G. Paterno: <u>CHAVEZ</u></p> <p>A. Materno: <u>FLORES</u></p> <p>Origen: <u>POTOSI</u></p> <p>F. Nacimiento: <u>25-12-94</u></p> <p>Edad: <u>12</u> Est:..... Peso:.....</p> <p style="text-align: center;"><i>Sello y Firma Comité Técnico</i></p>	<p style="text-align: center;">II. CAMPEONATO NACIONAL DE FÚTBOL INFANTIL SUB 13 "EL ALTO - BOLIVIA 2006" Reg. N°</p>  <p>Nombres: <u>MISHAEL</u></p> <p>H. Paterno: <u>CHAVEZ</u></p> <p>A. Materno: <u>PAGE</u></p> <p>Origen: <u>POTOSI</u></p> <p>F. Nacimiento: <u>1-01-94</u></p> <p>Edad: <u>12</u> Est:..... Peso:.....</p> <p style="text-align: center;"><i>Sello y Firma Comité Técnico</i></p>

B.4. Ficha Inscripción Deportista

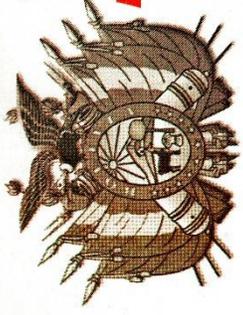
FUNDACION ADEFUSA BOLIVIA

MAT.2012 - F01

FICHA DE INSCRIPCION MATRIZ

FECHA DE INSCRIPCION	MES		DIA		AÑO	
APELLIDO PATERNO						
APELLIDO MATERNO						
NOMBRES						
FECHA DE NACIMIENTO	MES		DIA		AÑO	
CARNET DE IDENTIDAD						
DEPARTAMENTO	LA PAZ					
PROVINCIA			CIUDAD			
DOMICILIO PERMANENTE			ZONA			
CATEGORIA						
TELEFONO CELULAR			COTEL			
EDAD ACTUAL						
UNIDAD EDUCATIVA			CURSO			
LUGAR DE LA UNIDAD EDUCATIVA						
PRIVADO						
PUBLICO						
CODIGO PERSONAL						
CODIGO DE LA FUNDACION						
NOMBRE DEL PADRE						
NOMBRE DE LA MADRE						
	<input type="checkbox"/>	FOTOCOPIA CARNET DE IDENTIDAD				
	<input type="checkbox"/>	FOTOCOPIA CERTIFICADO NACIMIENTO				
	<input type="checkbox"/>	FOTOCOPIA DE LIBRETA ESCOLAR				
	<input type="checkbox"/>	AUTORIZACION DEL PADRE				
	<input type="checkbox"/>	2 FOTOS 3X3 - FONDO ROJO				
	<input type="checkbox"/>	FICHA MEDICA				
	<input type="checkbox"/>	FICHA DENTAL				
	<input type="checkbox"/>	SOCIAL				
	<input type="checkbox"/>	CULTURA				
	<input type="checkbox"/>	DEPORTE				
	<input type="checkbox"/>	EDUCATIVA				
	<input type="checkbox"/>	SALUD				
OBSERVACIONES:						
RESPONSABLE DEL PROYECTO:	DIRECTOR DEL PROYECTO					

B.5. Libreta Calificación Deportista



GOBIERNO MUNICIPAL DE LA CIUDAD DE EL ALTO
DIRECCIÓN MUNICIPAL DE DEPORTES
LIBRETA DE CALIFICACIONES DEPORTIVAS



FOTO

No. 027 Año 200 4

APELLIDOS Flores Buisne FECHA DE NACIMIENTO 15/05/1998 C.I. PESO

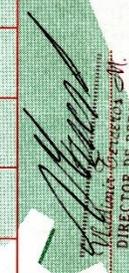
NOMBRES JOEL JOSUE FECHA DE INCORPORACION 19/04/2004 ESTATURA

CATEGORIA SUB-11 DISCIPLINA FUTBOL DIRECCION ALPACARITA

PERIODO	TECNICA	PREPARACION FISICA	SALUD	CAPACITACION TEORICA	COMPANERISMO	TEMPERAMENTO	ASISTENCIA	CONDUCTA	HIGIENE	OBSERVACIONES	FIRMA DEL PROFESOR	FIRMA DEL PARTICIPANTE	FIRMA DEL PADRE, MADRE O TUTOR
ENERO													
FEBRERO													
MARZO													
ABRIL	4	4	5	4	5	5	4	5	4				
MAYO	4	4	5	4	5	6	4	5	4				
JUNIO	4	4	5	4	5	5	5	5	4				
JULIO	5	5	6	5	6	5	6	5	4				
AGOSTO	5	6	6	5	6	5	6	5	4				
SEPTIEMBRE	5	6	6	5	6	6	4	5	5				
OCTUBRE	5	5	6	5	6	6	4	4	5				
NOVIEMBRE	5	6	6	5	6	5	4	4	6				
DICIEMBRE	5	5	6	5	6	6	4	4	6				
PROMEDIO	4.6	5	5.6	4.6	5.5	5.4	4.5	4.6	4.7				

Promovido a la División Superior de:

Dejando en Libertad el:



DIRECTOR DE DEPORTES
GOBIERNO MUNICIPAL DE EL ALTO



VICE - PRESIDENTE
A.F.E.-A.

Anexo C

C.1. Manual de Usuario

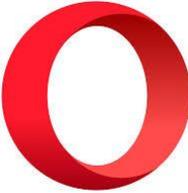
MANUAL DEL USUARIO

INTRODUCCIÓN.

El Sistema se ha diseñado para entorno Web, la cual fue diseñado con herramientas orientadas a la plataforma web, este sistema puede ser mediante navegadores de internet escribiendo la dirección web asignada.

REQUISITOS TÉCNICOS

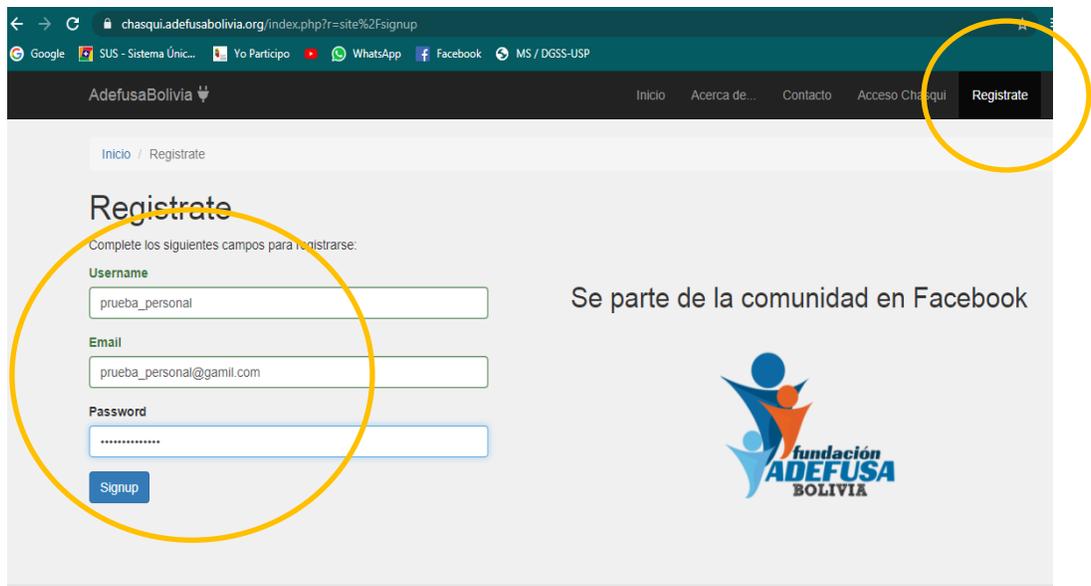
- a) Navegador compatible con el Sistema. Los navegadores soportados por el sistema son:

		
GOOGLE CHROME	OPERA	MICROSOFT EDGE

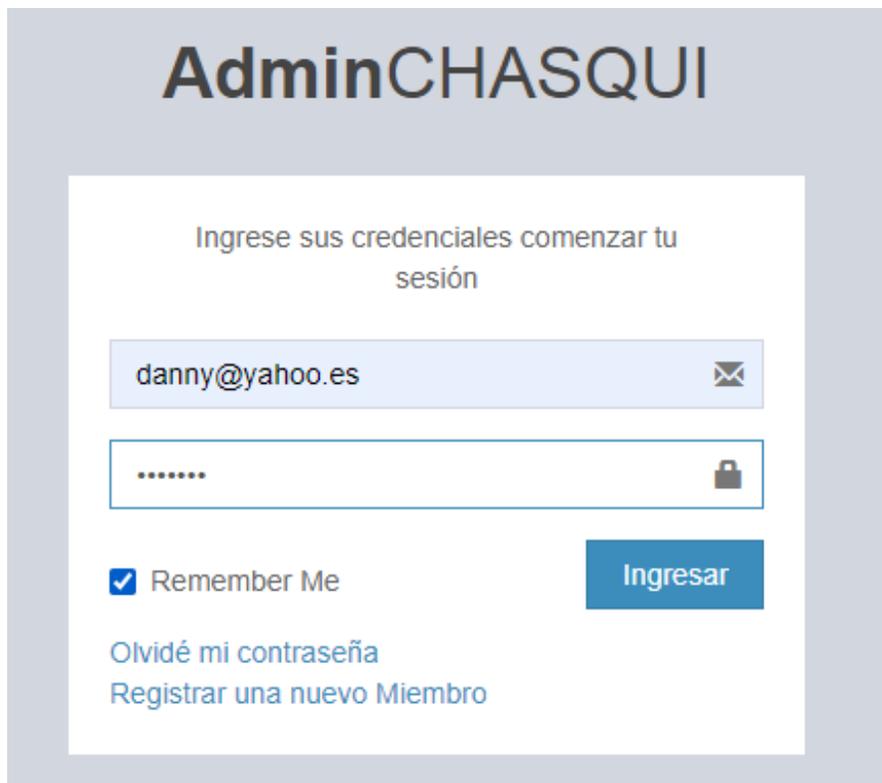
- b) Acceso Internet.
c) Estar Registrado en el sistema, con un rol dentro el sistema, otorgando un usuario y contraseña.

REGISTRO Y ACCESO AL SISTEMA

- a) Registro al sistema.** Previamente el usuario debe registrarse al sistema ingresando las credenciales que solicite al usuario. Para ello debe dirigirse al siguiente link: **chasqui.edefusabolivia.org**



- 1) **Debe ir al menú regístrate**, esto es una sola vez.
 - 2) **Debe llenar los campos de manera obligatoria**, y esperara que el administrador otorgue los permisos necesarios de acuerdo a su rol y tipo de usuario que le corresponda.
- b) **Acceso al Sistema.** Para poder acceder al sistema, debe dirigirse al siguiente enlace: **admin.edefusabolvia.org**

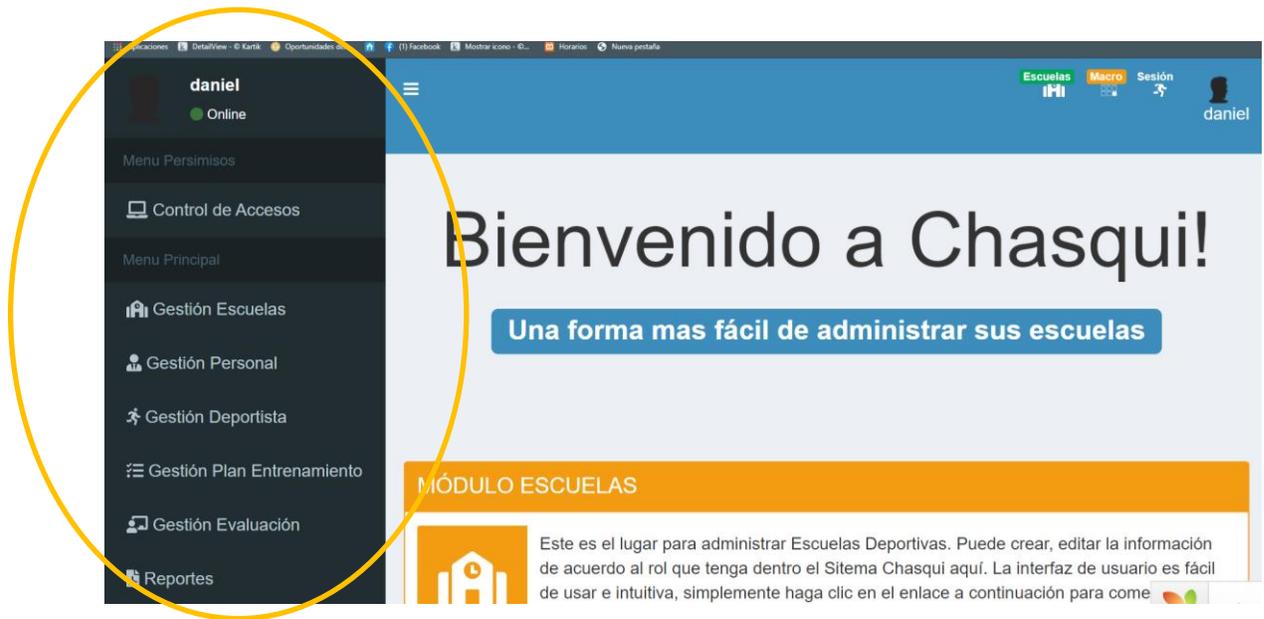


Debe ingresar el correo y la contraseña que registró para poder ser dado de alta en el sistema.

MANUAL DEL ADMINISTRADOR DEL SISTEMA

INTRODUCCIÓN.

En la cuenta administrados de sistema, vera la siguiente ventana



CONTROL DE ACCESOS

Solo podrá acceder al menú control de accesos los administradores del sistema, con privilegio de SUPERADMIN y tendrá el control total del sistema.

Las tareas que pueden realizar son:

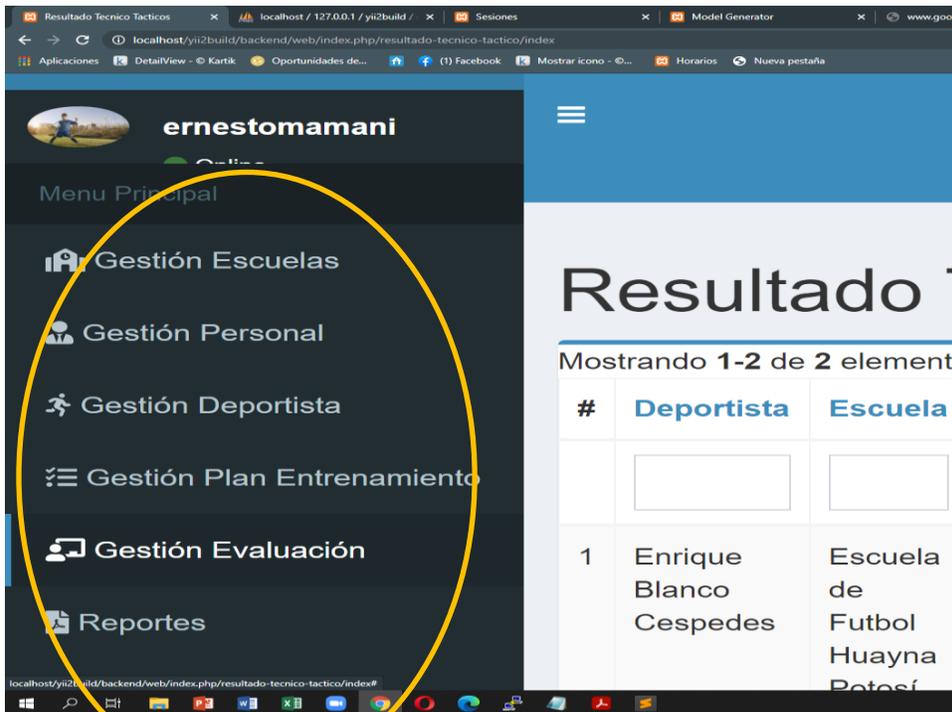
CREAR		
ROLES	siempre y cuando no sea mayores al Super Admin	<40

<p>TIPOS DE USUARIO</p>	<p>Debe tener en cuenta, ésta debe cumplir una función dentro del sistema. Debe actualizar en sistema dicho usuario. Recomendable preguntar al Administrador del sistema</p>	<p>Debe tener conocimiento el desarrollador del sistema</p>
<p>ESTADO DEL USUARIO</p>	<p>Normalmente son tres estados ACTIVO, INACTIVO Y PENDIENTE</p>	<p>Valor por defecto Activo</p>
<p>DATOS DE INICIALIZACIÓN</p>	<p>En caso de migración de datos, aquí debe inicializar los datos queridos para que el sistema pueda trabajar correctamente.</p>	<p>Se recomienda realizar backups del sistema</p>
<p>ACTUALIZAR</p>		
<p>USUARIO</p>	<p>Puede actualizar, el estado del usuario, el rol, tipo de usuario, nombre de usuario y correo</p>	<p>Tome debida precaución ya que la actualización hará que cambie datos que hubiera generado el usuario.</p>

PERFIL DEL USUARIO	Puede corregir, cambiar algún dato erróneo del usuario	Debe consultar con el usuario antes de realizar cambios
ELIMINAR		
<p>Se recomienda no eliminar ningún dato, a menos que esté completamente seguro de que no tiene datos relacionados con el ítem a eliminar. Consulte a su administrador del sistema para mayor información.</p>		

MANUAL DEL USUARIO DEL SISTEMA

INTRODUCCIÓN. En la cuenta Coordinador de sistema, vera la siguiente ventana



Tiene el acceso de los siguientes módulos:

a) **Gestión Escuelas**

Las acciones autorizadas con las siguientes:



- *Campo Deportivo.* Crear, actualizar, eliminar, ver e imprimir reporte, una categoría de campo deportivo
- *Sede Deportiva.* Crear, actualizar, eliminar, ver e imprimir reporte, una sede deportiva
- *Disciplina Deportiva.* Crear, actualizar, eliminar, ver e imprimir reporte.
- *Escuela Deportiva.* Crear, actualizar, eliminar, ver e imprimir reporte.
- *Asignar Horario a Escuela Deportiva.* Crear, actualizar, eliminar, ver e imprimir reporte.

SEDE DEPORTIVA

#	Nombre	Tipo	Distrito	Dirección	Foto
1	SEDE DEPORTIVA CIUDAD SATÉLITE	B	DM-3	ZONA CALLUYO	
2	SEDE DEPORTIVA CIUDAD SATÉLITE	A	DM-1	AV. SATELITE, ZONA CIUDAD SATÉLITE NRO 5	

Detalle Sede

Sede Deportiva Ciudad Satélite

Actualizar Eliminar Listar

Disciplina Deportiva

Sede Deportiva Ciudad Satélite

Nombre: SEDE DEPORTIVA CIUDAD SATÉLITE
Tipo: CÉSPED CAMERINOS Y GRADERÍAS
Distrito: DM-1
Dirección: AV. SATELITE, ZONA CIUDAD SATÉLITE NRO 5

Actualizar Foto

Escuelas Deportivas

Sede Deportiva Ciudad Satélite

Total 3 elementos.

Escuela	Nuevo
ESCUELA DE FUTBOL SATÉLITE MAÑANA	
ESCUELA DE BASQUETBOL SATÉLITE TARDE	
ESCUELA DE RAQUET FRONTÓN SATÉLITE	

1. El contenedor del modulo Escuelas. Muestra las acciones que uno puede realizar dentro de este módulo.
 - Campo deportivo, Crear una categoria de campo deportivo, según catalogación por cada asociación deportiva.

- Sede Deportiva, Crear una sede deportiva donde trabajará cada disciplina deportiva.
 - Escuela Deportiva, La escuela deportiva específica, sea de futbol, Voleivol, según tenga la capacidad de la sede.
 - Horario Escuelas, Aquí puedes asignar un horario a una escuela determinada.
2. Muestra un listado de filas creadas de acuerdo a la opción requerida, muestra un resumen de los datos generados en dicha acción.
 3. Acciones de Filas, estas pueden mostrarse de acuerdo al rol del usuario, normalmente es VER, ACTUALILZAR, ELIMINAR Y EMPRIMIR
 4. Crear, es la accion para crear una nueva fila dentro de la base de datos, ingresando los datos que requieren

- a. En el formulario debe llenar los datos requeridos en especial aquel que está con (*) es obligatorio.
 - b. Debe seleccionar una imagen, con no mayor a 700*700px, jpg, png.
 - c. Una vez llenada todos los campos presione guardar y tendrá en la lista la nueva fila generada.
5. Imprimir, toda la lista generada.



ESCUELAS DEPORTIVAS

Adefusa Bolivia
Calle 22, nro. 343,
Huayna Potosí, El Alto
(591) 705-06082
info@adefusabolivia.org

FECHA DE IMPRESIÓN 24-Nov-2020

NOMBRE	SEDE	DEPORTE	FECHA DE CREACIÓN
ESCUELA DE FUTBOL HUAYNA POTOSÍ MAÑANA	ZONA DEPORTIVA HUAYNA POTOSÍ	FUTBOL	20-Oct-2020
ESCUELA DE FUTSAL VILLA INGENIO MAÑANA	ESTADIO VILLA INGENIO	FUTBOL SALA	20-Oct-2020
ESCUELA DE FUTBOL SATELITE MAÑANA	SEDE DEPORTIVA CIUDAD SATELITE	FUTBOL	29-Oct-2020
ESCUELA DE BASQUETBOL SATELITE TARDE	SEDE DEPORTIVA CIUDAD SATELITE	BASQUETBOL	29-Oct-2020
ESCUELA DE RAQUET FRONTON SATELITE	SEDE DEPORTIVA CIUDAD SATELITE	RAQUETA FRONTON	29-Oct-2020

NOTA:
Datos extraídos de la Base de Datos CHASQUI.

6. Detalle Vista, muestra un detalle de la fila seleccionada, en esta puede ACTUALIZAR, ELIMINAR, LISTAR.

Detalle Escuela x

Actualizar Escuela: Sede Deportiva Ciudad Satélite

Ingrese Datos (*) son obligatorios...

Sede Deportiva *
 x v

Nombre Escuela *

Deporte *
 x v

Guardar

b) Gestión Personal



- *Institución.* Crear, actualizar, eliminar, ver e imprimir reporte.
- *Cargo Personal.* Crear, actualizar, eliminar, ver e imprimir reporte.
- *Información Personal.* Crear nuevo personal, actualizar datos del personal, eliminar, ver e imprimir reporte.
- *Asignar al personal un Horario.* Crear, actualizar, eliminar, ver e imprimir reporte.

c) Gestión Deportista



- *Deportista.* Crear, actualizar, eliminar, ver e imprimir reporte.

1. Se debe llenar los campos requeridos, en especial aquellos que son obligatorios(*)
2. Arrastre o seleccione foto del personal no mayor a 700*700px, formato jpg, png.
3. Una vez llenada debe guardar los datos.

- *Asignar al Deportista un Horario.* Crear, actualizar, eliminar, ver e imprimir reporte.
- *Kardex Deportista.* ver e imprimir reporte

d) Gestión plan de entrenamiento



- *Plan de Entrenamiento*, ver e imprimir reporte.

Plan Entrenamientos

Crear Plan Entrenamiento 1

Mostrando 1-2 de 2 elementos.

#	Objetivo	Categoría	Género	Nro. Macro	Escuela	Personal	
1	participar en campeonato sub 13 AFLP	MOSQUITOS	Masculino 2	1	Escuela de Futbol Huayna Potosi Mañana	Rocio Gladis Mamani Mamani	3
2	Campeonato AFEAL	Mosquitos	Masculino	1	Escuela de futbol satélite mañana	Rocio Gladis Mamani Mamani	

1. *Crear plan de entrenamiento*, solo podrá realizar esta acción con el rol de entrenador.

Escuelas Designadas

Escuelas Designadas

Total 1 elemento.

#	Sede	Escuela	Horario
1	Zona Deportiva Huayna Potosí	Escuela de Fútbol Huayna Potosí Mañana	Lunes, Jueves, Sabado

Ingrese Datos (*) son obligatorios...

Objetivo *

Nro. Macro *

Objetivo no puede estar vacío.

Categoría *

Género *

Escuela *

Elija Una Categoría ...

Elija Género ...

Elija Escuela ...

Obs

Guardar

- Muestra una lista de Escuelas Designadas
- Llenar los campos requeridos.
- Guardar.

- Muestra una lista de planes de entrenamiento generados
- Acciones permitidas según el rol del usuario

participar en campeonato sub 13 AFLP

Actualizar Eliminar Listar

Nro. Macro 1

Objetivo participar en campeonato sub 13 AFLP

Categoría MOSQUITOS

Género Masculino

Escuela Escuela de Fútbol Huayna Potosí Mañana

Personal Rocio Gladis Mamani Mamani

Obs

Fecha Creada Nov 8, 2020

Periodos Mesociclo Micro - Ciclos Fecha - Sesiones Sesiones - Programadas

Periodos

Total 1 elemento.

Periodo	Descripcion	Nuevo
Preparatorio - General		

En la acción Vista Detalle

- a. Puede, realizar las acciones permitidas según el rol del usuario.
- b. Las acciones permitidas son normalmente permitidas, para el llenado de MacroCiclo correspondiente.
- c. Puede Crear, eliminar o actualizar, según requiera.

Crear Programacion +

(*) es obligatorio

Plan *

 a

Nro. de Sesión *

Objetivo Sesión *

 b

Escuela *

Microciclo *

Fecha *

Select...

Select...

 2020-11-02 **c**

- a. Debe llenar los campos, que solicita, el Plan de trabajo, la escuela, el número de sesión,
- b. la sesión, que muestra el objetivo a alcanzar en la sesión, el Microciclo correspondiente (MES)
- c. Importante la fecha a programar, dato eso datos el sistema programara a todos sus deportistas para la fecha programada como NR.

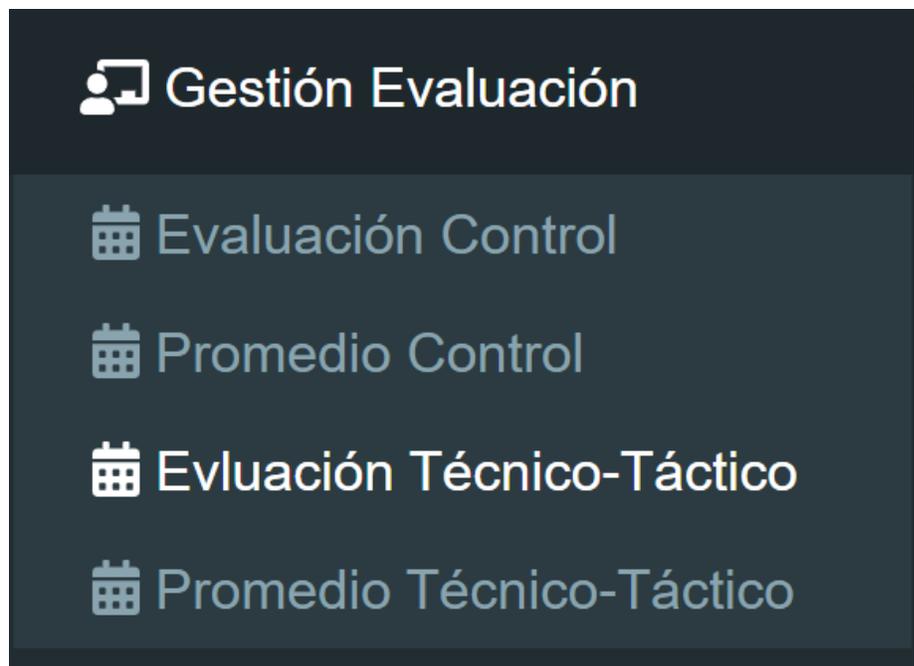
Mostrando 14 de 4 elementos.

#	Número Sesión	Escuela	Deportista	Fecha	Asistencia
1	2	Escuela de Futbol Huayna Potosi Mañana	juan pedro alabana pamplona	Nov 2, 2020	NR c
2	1	Escuela de Futbol Huayna Pot Mañana	juan pedro alabana pamplona	Nov 2, 2020	P b
3	1	Escuela de Futbol Huayna Pot Mañana	della martha andaluz monasterios	Nov 2, 2020	P
4	1	Escuela de Futbol Huayna Potosi Mañana	mendoza alavez monasterios	Nov 2, 2020	P

a

- a.** Puede dirigirse a la pestaña programación deportista, para ver sus programaciones realizadas y sesiones acabadas.
- b.** Es la lista presentada por fecha
- c.** Puede cambiar el estado de las programaciones la cual por defecto será:
 - NR: No registra, es cual crea una sesión programada.
 - L: Licencia o permiso del deportista.
 - P: Presente.
 - A: Ausente, no justificada.

e) Gestión Evaluación



- *Evaluación.* Crear, actualizar, ver e imprimir reporte de acuerdo al rol asignado.

- Promedio, ver e imprimir reporte.

Promedio Control Técnico

mendoza alavez monasterios

Listar

Deportista	mendoza alavez monasterios
Escuela	Escuela de Futbol Huayna Potosí Mañana
Entrenador	Rocio Gladis Mamani Mamani
Email	rociomamani@gmail.com
Fecha Evaluación	Oct 14, 2020



Resultado Promedial

PROMEDIO CONTROL DEL BALÓN	PROMEDIO CONTROL ELEVADO	PROMEDIO CONTROL-CONDUCCIÓN	PROMEDIO CONTROL ZIG-ZAG
★ ★ ★ ★ ★	★ ★ ★ ★ ★	★ ★ ★ ★ ★	★ ★ ★ ★ ★
Regular	En Aprendizaje	Regular	En Aprendizaje

Anexo D

Manual de Instalación

SISTEMA DE GESTION, CONTROL Y SEGUIMIENTO TÉCNICO PARA ESCUELAS DEPORTIVAS DE LA CIUDAD ALTO

CASO: ESCUELAS DEPORTIVAS DE FUTBOL FUNDACION ADEFUSA

ESPECIFICACIONES PARA INSTALACIÓN DEL SISTEMA (CHASQUI)

Versión 1.0

ESPECIFICACIÓN DE INSTALACIÓN DEL SISTEMA

Modo Producción

1. INTRODUCCION

1.1. Propósito

El propósito de este documento es para describir en forma detallada, los pasos a seguir para la instalación del Sistema de Control y Seguimiento Técnico para Escuelas Deportivas.

1.2. Alcance

Describe los requerimientos mínimos de la instalación en modo producción.

1.3. Definiciones, Acrónimos y Abreviaturas

Ver Glosario

1.4. Referencias

Sin Referencia

2. REQUERIMIENTO DEL SISTEMA

2.1. Breve Descripción

En el siguiente cuadro detalla los requisitos de la instalación de una Framework, cas particular Yii2, que muestran los requisitos mínimos de hardware y software para la instalación del Sistema desarrollado.

FICHA DE INSTALACIÓN DE FRAMEWORK PHP		
Rol responsable:	Administrador	
Nombre del framework:	Yii	
URL de descarga:	http://www.yiiframework.com/	
Versión:	2.0	
URL de instalación:	http://www.yiiframework.com/doc-2.0/guide-start-installation.html	
Requisitos de instalación:	Software:	PHP 5.4.0 o superior, apache última versión, un motor de base de datos
	Hardware:	Sistema básico / servidor web de contrato con licencias SSL y SSH.
Opciones o servicios que permite	<ul style="list-style-type: none">- Implementa el modelo MVC (Model-View-Controller)- Yii tiene la filosofía de que el código debe ser escrito de una manera sencilla pero elegante. Yii nunca tratará de sobrecargar cosas principalmente con el propósito de seguir estrictamente cierto patrón de diseño- Proporciona generadores de consultas y ActiveRecord para ambas bases de datos relacionales y NoSQL- Soporte de almacenamiento en caché de varios niveles- Permite crear extensiones para distribuir- Proporciona alto rendimiento	
Instrucciones de instalación:	<ol style="list-style-type: none">1. Crear una cuenta en https://github.com/2. Descargar composer. de https://getcomposer.org/Composer-Setup.exe e instalarlo	

	<p>3. Los comandos 4 y 5 deben ejecutarse en una consola de símbolo de sistema y debe estar ubicado en el directorio webRoot</p> <p>4. Instalar plugin de composer con el siguiente comando:</p> <pre>composer global require "fxp/composer-asset-plugin:1.0.0"</pre> <p>5. Crear el primer sitio con yii a través de composer, ejecutando el siguiente comando. El proyecto se creará en la carpeta basic del directorio htdocs de xampp. El comando es el siguiente:</p> <pre>composer create-project --prefer-dist yiisoft/yii2-app-basic basic</pre> <p>6. Después de la instalación con un navegador se puede verificar el proyecto Yii construido. La URL de prueba es la siguiente:</p> <pre>http://localhost/basic/web/index.php</pre> <p>7. Si en pantalla del navegador aparece la aplicación, felicitaciones ya puedes trabajar con Yii</p>
Dificultades en la instalación:	Confusión en el método de instalar el framework porque existen dos maneras, en esta ficha se dan las instrucciones para instalar Yii vía Composer

3. REQUISITOS MÍNIMOS DE HARDWARE

3.1 Servidor Local.

- a) Sistema Operativo, Windows (32/64 bits), Linux (32/64 Bits).
- b) Memoria RAM 2GB (32bits), 2GB o Memoria RAM de 4GB (64bits).
- c) Espacio Disponible de 16GB o superior.
- d) Procesador 1Gigaherzio (GHz) o superior.
- e) Controladores de Sistema Operativo debidamente instalados.
- f) Servidor Virtual (Wampp, Wampp)
- g) Apache 7.4 o superior.

h) Mysql-Server Versión del servidor: 10.4.11-MariaDB o superior.

3.1 Servidor Web.

a) Protección DDos, protección antivirus, autenticación de dos factores.

b) Versión de PHP 7.4 o superior, acceso SSH, Mysql/MariaDB.

c) Subdominios, acceso FTP, almacenamiento SSD mínimo de 15BG

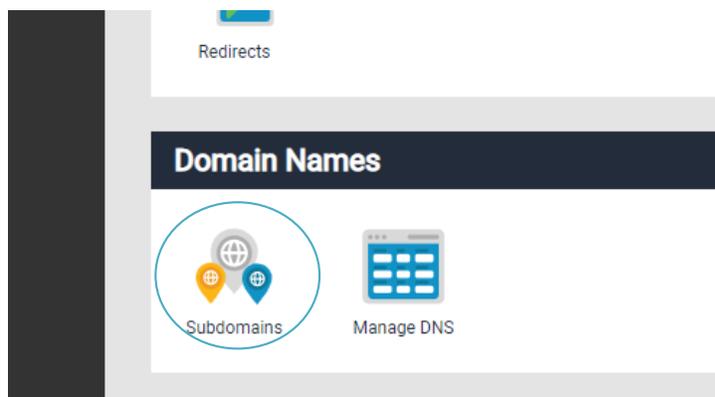
d) Transferencia Ilimitada, Administrador de Backup.

4. CONFIGURACIÓN DE SUB-DOMINIOS

a) Crear dos sub-dominios:

Backend: admin.edefusabolvia.org

Frontend: chasqui.edefusabolvia.org



admin.edefusabolvia.org

chasqui/backend/web/

Domain will point to:/home/sites/3b/f/f8a35aca87/chasqui/backend/web/

chasqui.edefusabolvia.org

chasqui/frontend/web/

Domain will point to:/home/sites/3b/f/f8a35aca87/chasqui/frontend/web/

La estructura de archivos se debe configurar como en la imagen anterior.

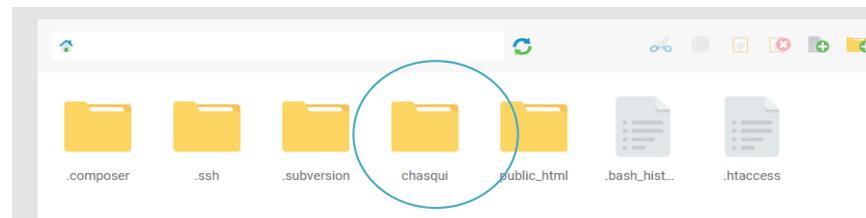
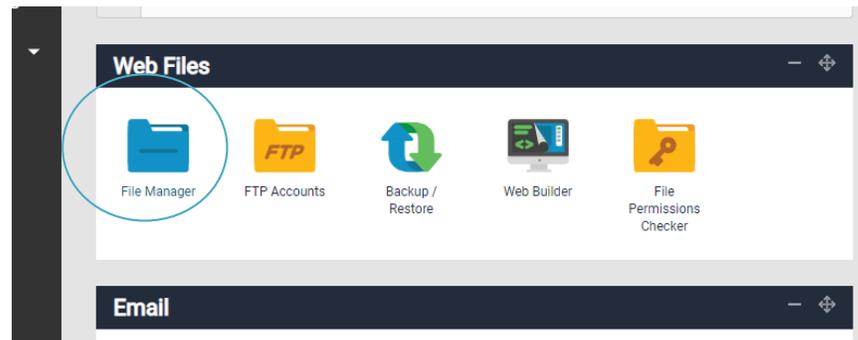
5. Transferir Archivos del Sistema

5.1. Modo Local

1. Descomprimir el archivo comprimido en:
2. Unidad C:xampp/htdocs/...
Descomprimir el dentro la carpeta raíz.

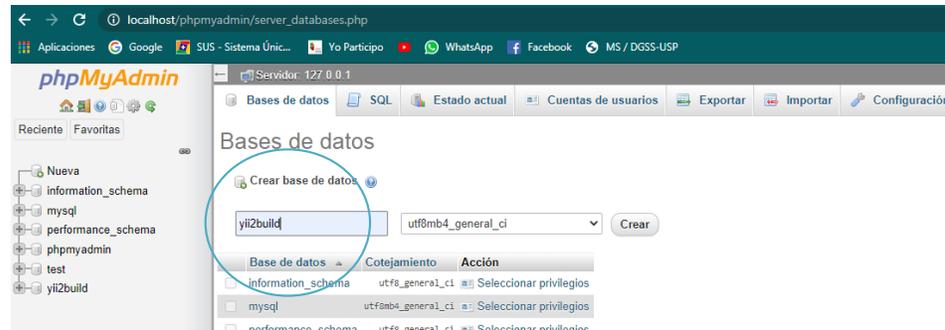
5.2. Modo Remoto

- a) Subir el sistema vía FTP, o descomprimir el archivo desde cPanel.

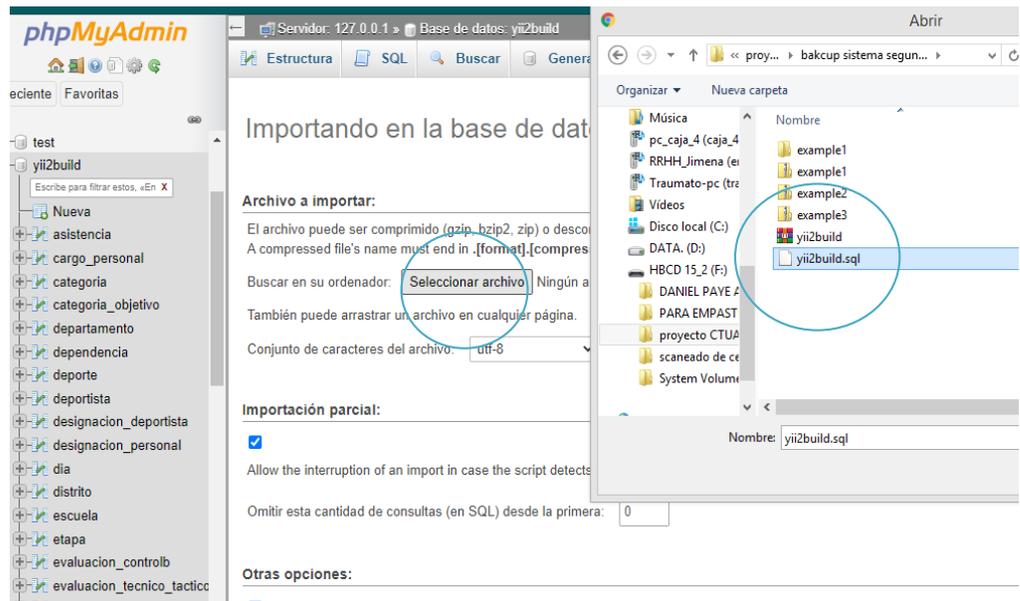


Crear una carpeta "chasqui", que es la carpeta raíz del sistema, descomprimir el archivo.

- b) Crear la Base de Datos yii2build en Mysql



c) Subir la estructura del BD.



d) Probar la instalación. Debe escribir la siguiente dirección, según sea el caso.

Modo

local:

<http://localhost/yii2build/backend/web/index.php/site/login>

Modo Web:

<https://admin.edefusabolivia.org/>

Se tiene un usuario administrador, para el inicio del sistema una vez ingresado al sistema. Puede otorgar un nuevo usuario los privilegios de Administrador (SuperAdmin), para ello debe crear un usuario desde el frontend

AdminCHASQUI

Ingrese sus credenciales comenzar tu sesión

danny@yahoo.es

Remember Me

Ingresar

[Olvidé mi contraseña](#)

[Registrar un nuevo Miembro](#)

USUARIO: danny@yahoo.es, CONTRASEÑA:*****

La puede pedir al momento de adquirir el software.

Anexo E

Avales de Conformidad

Ciudad de El Alto, noviembre de 2020

Señor:

Ing. David Carlos Mamani Quispe

DIRECTOR CARRERA INGENIERÍA DE SISTEMAS – UPEA

Presente.-

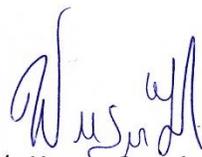
REF.: AVAL DE CONFORMIDAD

Distinguido(a) ingeniero(a):

*Tengo a bien dirigirme a su persona para comunicarle mi conformidad del trabajo final del Proyecto de Grado, “**SISTEMA DE GESTIÓN Y SEGUIMIENTO TÉCNICO PARA ESCUELAS DEPORTIVAS DE EL ALTO**” CASO: **ESCUELAS DEPORTIVAS DE FUTBOL FUNDACIÓN ADEFUSA**, elaborado por el universitario **DANIEL FLORENCIO PAYE ARUQUIPA**, con cedula de identidad **4988154L.P.** y **R.U.:200365**, para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.*

Sin otro particular, me despido de usted.

Atentamente.



Ing. Wendy Yomar Sarmiento Martínez

TUTOR REVISOR

Ciudad de El Alto, noviembre de 2020

Señor:

Ing. David Carlos Mamani Quispe

DIRECTOR CARRERA INGENIERÍA DE SISTEMAS – UPEA

Presente.-

REF.: AVAL DE CONFORMIDAD

Distinguido ingeniero:

Tengo a bien dirigirme a su persona para comunicarle mi conformidad del trabajo final del Proyecto de Grado, titulado **“SISTEMA DE GESTIÓN Y SEGUIMIENTO TÉCNICO PARA ESCUELAS DEPORTIVAS DE EL ALTO” CASO: ESCUELAS DEPORTIVAS DE FUTBOL FUNDACIÓN ADEFUSA**, elaborado por el universitario **DANIEL FLORENCIO PAYE ARUQUIPA**, con cedula de identidad **4988154L.P.** y **R.U.:200365**, para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.

Atentamente.


Ing. Sergio Ramiro Rojas Saire
TUTOR ESPECIALISTA

Ciudad de El Alto, noviembre de 2020

Señor:

Ing. David Carlos Mamani Quispe

DIRECTOR CARRERA INGENIERÍA DE SISTEMAS – UPEA

Presente.-

REF.: AVAL DE CONFORMIDAD

Distinguido ingeniero:

Tengo a bien dirigirme a su persona para comunicarle mi conformidad del trabajo final del Proyecto de Grado, titulado **“SISTEMA DE GESTIÓN Y SEGUIMIENTO TÉCNICO PARA ESCUELAS DEPORTIVAS DE EL ALTO” CASO: ESCUELAS DEPORTIVAS DE FUTBOL FUNDACIÓN ADEFUSA**, elaborado por el universitario **DANIEL FLORENCIO PAYE ARUQUIPA**, con cedula de identidad **4988154L.P.** y **R.U.:200365**, para su defensa pública y evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido de usted.

Atentamente.



Ing. Marisol Arguedas Balladares

TUTOR METODOLÓGICO