

Universidad de las Ciencias Informáticas

Facultad 8



**Análisis, diseño e implementación del módulo pruebas
médicas del sistema de planificación del entrenamiento
del judo femenino**

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

Autores:

Omar Mora Pérez

Francisco Rodríguez González

Tutoras:

Ing. Aliuska Sánchez Ibarria

Ing. Karenia Donatien Goliath

Ciudad de La Habana, mayo del 2010

“Año 52 de la Revolución”

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo que lleva por título “Análisis, diseño e implementación del módulo pruebas médicas del sistema de planificación del entrenamiento del judo femenino”, y autorizamos a la Universidad de las Ciencias Informáticas (UCI) a que haga el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del 2010.

Autores:

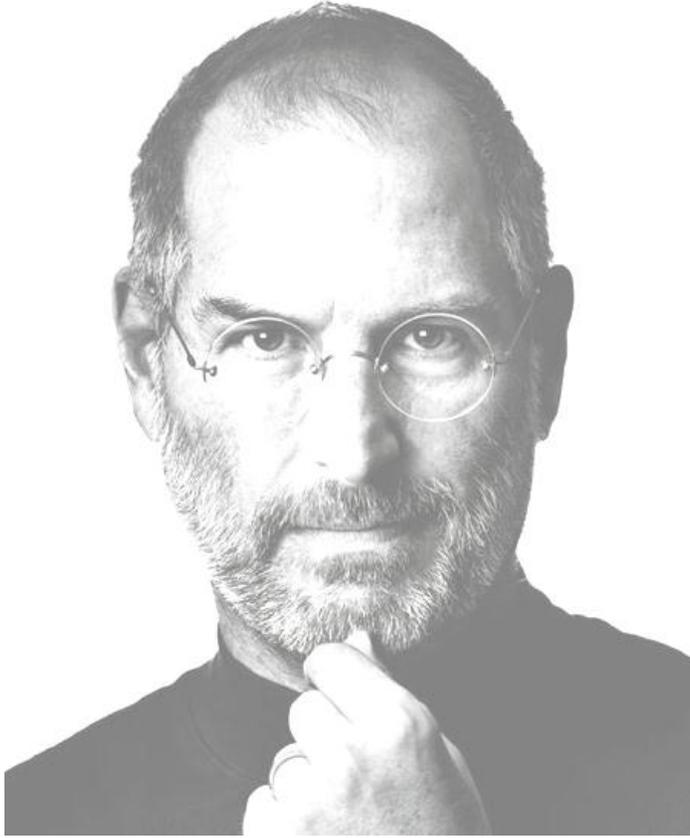
Omar Mora Pérez

Francisco Rodríguez González

Tutoras:

Ing. Karenia Donatien Goliath

Ing. Aliuska Sánchez Ibarria



“El único modo de hacer un gran trabajo es amar lo que haces. Si no lo has encontrado todavía, sigue buscando. No te acomodes. Como con todo lo que es propio del corazón, lo sabrás cuando lo encuentres.”

Steve Jobs

Dedicatoria

Francisco

A mi mamá por estar tan orgullosa de mí, por todo su amor, dedicación, apoyo y por la ternura que siempre me profesa.

A mi papá por ser mi amigo, por su eterno amor, por su confianza y por ser mi inspiración para lograr lo que soy.

A mis hermanos que me han llenado de alegrías, ojala algún día mi camino les sirva para recorrer el suyo.

A mis abuelos, que tanta ternura han puesto en esta semilla para que diera sus frutos.

Omar

A mis hermanos por ser los que siempre me han apoyado directamente.

Agradecimientos

Francisco

A mis padres por confiar siempre en mí, por estar a mi lado en cada momento de mi vida, por sus enseñanzas, sus consejos, su comprensión y respetar mis decisiones, por su sacrificio y su infinito cariño.

A mi familia que tanto apoyo me dedicaron, a mis abuelitas Yoyi y Silvia, a mis abuelos Paquito (de Pueblo Viejo) y Paquito (del Batey), a Maye, a mis tíos y primos.

A Yadira que desde lejos siempre comparte mis buenos y malos momentos, y siempre está ahí.

A Yudita que me alentó y ayudó incansablemente en los momentos que más lo necesitaba.

A mis amistades que tanto han significado en mi vida en esta escuela, a los que están y los que no. A los del aula, a los de cuarto, a los que siguen conmigo desde el pre.

A mis tutoras por el esfuerzo y la paciencia que depositaron en nosotros.

A esta Revolución que me dio la posibilidad de estudiar en una universidad como esta.

A Fidel por ser el máximo ejemplo y guía de nuestra Revolución.

A todas esas personas que entraron en mi vida y me han ayudado a ser quien soy...

Muchas Gracias...

Omar

A mi sobrino por ayudarme a salir de momentos difíciles y apoyarme incondicionalmente.

A Maite por creer en mí y dar todo lo mejor de sí para apoyar mi objetivo.

A mis padres, que de no ser por ellos no hubiese logrado lo que soy ahora.

A mis amigos Sori, Melquiades y Yoenni por estar en todos los momentos, tanto malos como buenos, cuidándome y enseñándome el camino a seguir.

A la Revolución por permitir que jóvenes de todas partes tengan la posibilidad de graduarse en una universidad como esta.

En fin, a todo aquel que de alguna manera ayudó a mi formación como profesional.

Muchas Gracias...

Resumen

El trabajo de diploma que lleva por título “Análisis, diseño e implementación del módulo pruebas médicas del sistema de planificación del entrenamiento del judo femenino” tiene como objetivo la implementación de una aplicación Web para la gestión de la información referente a las pruebas médicas realizadas a las atletas en la Escuela Cubana de Judo Femenino. El desarrollo de esta aplicación estuvo guiado por los principios y reglas que propone la metodología XP, obteniendo los artefactos propuestos en cada iteración. Se presentó el estudio e investigación realizados sobre las diferentes tecnologías y herramientas que se utilizan para la creación de aplicaciones Web, además de las pruebas realizadas al sistema para garantizar su calidad y aceptación por parte del cliente.

Estructuración del contenido.

Capítulo 1: Fundamentación teórica.

Se expone la base teórica sobre la que se fundamenta el sistema para la gestión de la información de las pruebas médicas del entrenamiento del judo femenino. Se efectuó un estudio de los sistemas similares a nivel nacional como internacional. Se realizó un estudio y selección de las metodologías, estándares para el desarrollo de software, tecnologías actuales, lenguajes de programación y herramientas que se utilizarán en el desarrollo del sistema.

Capítulo 2: Desarrollo de la solución.

Se definieron los procesos de negocio y se elaboró una propuesta de sistema, además de la documentación de los diferentes artefactos según los flujos de trabajo de la metodología XP.

Capítulo 3: Implementación y pruebas del sistema.

Se explicaron los temas relacionados con la implementación del sistema.

Índice

Introducción	1
Capítulo 1: Fundamentación teórica	3
1.1 Introducción.....	3
1.2 Fundamentación del tema	3
1.3 Sistemas de gestión y planificación de pruebas médicas	4
1.3.1 Sistemas similares a nivel internacional	4
1.3.2 Sistemas similares a nivel nacional	5
1.3.3 Resultados del estudio de los sistemas similares analizados	5
1.4 Estudio de las metodologías y estándares para el desarrollo de software	6
1.4.1 Metodologías de desarrollo de software	6
1.4.2 Lenguaje Unificado de Modelado (UML).....	12
1.5 Tendencias y tecnologías actuales. Selección de las herramientas y lenguajes de desarrollo.	13
1.5.1 Lenguajes de programación	14
1.5.2 Entorno de desarrollo integrado (IDE).	19
1.5.3 Herramientas CASE de Modelado con UML.....	20
1.5.4 Sistema Gestor de Bases de Datos (SGBD)	22
1.6 Conclusiones.....	25
Capítulo 2: Desarrollo de la solución	26
2.1 Introducción.....	26

2.2 Descripción de las acciones vinculadas al campo de acción	26
2.2.1 Flujo actual de eventos	26
2.2.2 Objetos de automatización	27
2.2.3 Propuesta del sistema	27
2.3 Desarrollo de la solución	28
2.3.1 Fases propuestas por la metodología.....	28
2.3.2 Personas relacionadas con el sistema.....	29
2.3.3 Historias de Usuario	30
2.3.4 Iteraciones	36
2.3.5 Estimación del esfuerzo por historia de usuario	37
2.4 Plan de Entrega.....	38
2.5 Conclusiones.....	39
Capítulo 3: Implementación y pruebas	40
3.1 Introducción.....	40
3.2 Diseño del sistema	40
3.2.1 Tarjetas CRC	40
3.2.2 Diagrama de Despliegue	44
3.2.3 Arquitectura del sistema	45
3.3 Implementación	47
3.3.1 Iteración 1	47
3.3.2 Iteración 2	48
3.3.3 Iteración 3	48

3.3.4 Tareas de desarrollo genéricas	49
3.4 Estándares de codificación	50
3.4.1 Definiciones generales.....	51
3.4.2 Comentarios	52
3.4.3 Sentencias	52
3.5 Pruebas.....	53
3.5.1 Pruebas de aceptación	54
3.6 Conclusiones.....	67
Conclusiones generales.....	68
Recomendaciones	69
Referencias Bibliográficas.....	70
Bibliografía	73
Glosario de términos	75

Introducción

Cuba es conocida mundialmente por la buena calidad de sus atletas y sus victorias en muchas disciplinas deportivas. Esto se debe en gran medida al entrenamiento y preparación que poseen los deportistas y a la entrega incondicional de los excelentes entrenadores con los que cuenta el Instituto Nacional de Deporte, Educación Física y Recreación (INDER), los cuales han desempeñado un magnífico papel en el rendimiento, preparación y logros de los atletas cubanos. Para lograr estos rendimientos muchas personas han puesto todo su empeño en el desarrollo de nuevas técnicas y herramientas que faciliten la labor de entrenadores, médicos y especialistas de deporte.

En el proceso de control de la preparación de los atletas es de vital importancia la aplicación de un conjunto de pruebas médicas, científicamente concebidas, aplicadas y dirigidas a diagnosticar las condiciones competitivas, así como la evaluación del estado de preparación para un evento determinado. Los resultados de la aplicación de las pruebas médicas permiten observar los incrementos lógicamente esperados en las capacidades del atleta, los cuales se toman como elemento indicador de la efectividad de la preparación deportiva.

La realización de las pruebas médicas en la actualidad es un proceso muy engorroso para los entrenadores de la Escuela Cubana de Judo Femenino, pues para su elaboración es necesaria la realización de una serie de cálculos y búsquedas de información, la cual se encuentra dispersa y no está digitalizada. Los cálculos que se realizan por parte del médico en muchos casos lleva la utilización de extensas fórmulas de alta complejidad que dificultan el trabajo manual, dan paso a errores y omisión de datos que contrastan con el buen desempeño de esta actividad. Actualmente existe un sistema para la planificación del entrenamiento de las judocas que aún no cuenta con un módulo para la planificación y control de las pruebas médicas, lo que hace que los médicos se vean sometidos a utilizar métodos obsoletos para el desempeño de su trabajo, restando calidad y eficiencia en los resultados de su labor.

Teniendo en cuenta la situación actual surge el **problema científico**: ¿Cómo facilitar el proceso de gestión de las pruebas médicas para la planificación del entrenamiento del judo femenino?

Por lo planteado anteriormente se deriva como **objeto de estudio**: Los procesos manuales e informatizados de gestión de pruebas médicas para el entrenamiento deportivo.

Enfocando la investigación hacia el **campo de acción** de la informatización de los procesos de gestión de las pruebas médicas para el entrenamiento del judo femenino en la Escuela Cubana de Judo Femenino.

El **objetivo general** de este trabajo consiste en crear el módulo de pruebas médicas del sistema de planificación del entrenamiento del judo femenino que permita la informatización de la gestión de las pruebas médicas dentro del entrenamiento.

Defendiendo como idea que si se desarrolla un sistema que resuelva todos los requerimientos funcionales y no funcionales, sin errores de implementación y que informaticice el proceso de gestión de las pruebas médicas complementando el sistema de planificación del entrenamiento de judo femenino, entonces se facilitará el trabajo de los entrenadores y médicos de la Escuela Cubana de Judo Femenino.

Los **objetivos específicos** a cumplir son:

1. Realizar el diseño teórico de la investigación.
2. Revisar y estudiar la existencia de otras aplicaciones o soluciones similares.
3. Seleccionar el lenguaje y metodología de desarrollo de software a utilizar.
4. Seleccionar las herramientas que se van a utilizar según la metodología seleccionada.
5. Realizar la captura de requisitos del sistema.
6. Elaborar la documentación de los diferentes artefactos según los flujos de trabajo que proponga la metodología seleccionada.
7. Implementar el software en dependencia de las funcionalidades que debe tener.

Como **métodos de investigación** se utilizaron el **analítico – sintético** con el objetivo de analizar las teorías y documentos permitiendo la extracción de los elementos más importantes que se relacionan con el proceso gestión de las pruebas médicas en el entrenamiento de las atletas en la Escuela Cubana de Judo Femenino que incluyen plantillas de mediciones, fórmulas y métodos que se utilizan en el sistema de planificación de las pruebas médicas. Con el apoyo del **análisis histórico – lógico** se llevó a cabo la investigación sobre las aplicaciones y sistemas informáticos similares implementados en Cuba y el resto del mundo. Para conocer cuáles son las necesidades de los entrenadores, médicos y personal técnico que llevan a cabo las actividades de planificación del entrenamiento deportivo en la Escuela Cubana de Judo Femenino, se realizaron **entrevistas**.

Capítulo 1: Fundamentación teórica

1.1 Introducción

En este capítulo se expone la base teórica sobre la que se fundamenta el módulo pruebas médicas del sistema de planificación del entrenamiento del judo femenino. Se investiga acerca de los sistemas similares que existen tanto a nivel nacional como internacional dentro del ámbito deportivo. Además se realiza un estudio de las metodologías, estándares para el desarrollo de software, tecnologías actuales, lenguajes de programación y herramientas, así como la selección de las adecuadas para usarlas en el desarrollo del sistema que dará solución a la problemática actual.

1.2 Fundamentación del tema

Las pruebas médicas son debidamente concebidas y desarrolladas por el personal de medicina deportiva (médicos, psicólogos y fisioterapeutas) a los deportistas en coordinación con el entrenador, donde se controla el estado de salud, se realizan mediciones antropométricas, pruebas de laboratorio y pruebas funcionales. Es más común su aplicación en atletas de alto rendimiento aunque son aplicables a cualquier nivel siempre que exista la posibilidad real de utilizarlos.

La planificación de las pruebas médicas para el entrenamiento de las judocas consiste en un plan que identifica todas las pruebas por las que deben pasar las atletas en un período determinado y definido por los entrenadores. Este plan tiene como objetivo mantener las óptimas condiciones competitivas en las atletas y tiene dependencia con el plan gráfico del entrenamiento que realizan.

Actualmente existe un sistema encargado de la planificación del entrenamiento deportivo de las judocas que no incluye los procesos de gestión y control de las pruebas médicas del entrenamiento entre sus funcionalidades. Los procesos antes mencionados se realizan de forma manual por parte de los médicos por lo que se les dificulta la recolección de los datos de las pruebas médicas aplicadas en los diferentes ciclos de entrenamiento, trayendo consigo la pérdida de datos y su difícil manipulación.

Con este trabajo se pretende informatizar el proceso de gestión de las pruebas médicas del entrenamiento de las judocas en la Escuela Cubana de Judo Femenino creando una aplicación que complemente el sistema de planificación del entrenamiento del judo femenino que está actualmente en uso.

1.3 Sistemas de gestión y planificación de pruebas médicas

En la actualidad existen múltiples sistemas que facilitan la planificación del entrenamiento deportivo y en algunos casos contemplan la gestión de las pruebas médicas en estos ciclos. Haciendo una investigación tanto a nivel nacional como internacional se identificaron los siguientes:

1.3.1 Sistemas similares a nivel internacional

X-Medalist: Es un software que se encarga de planificar y controlar el entrenamiento para deportes individuales. Incorpora una herramienta que se va actualizando a medida que se utiliza el plan de entrenamiento, permite trabajar de forma rápida y coordinada, organizando toda la actividad laboral. En la planificación, permite incorporar una ilimitada cantidad de planes de entrenamiento por cada persona que se encuentre registrada dentro del sistema. También permite configurar todos los objetivos que se necesiten para la planificación y también en función del deporte que se trate. Permite llevar un registro y control de lesiones, historia clínica y deportiva y del análisis de la técnica individual y el biorritmo personal. Dispone de una herramienta que le permite comparar volúmenes de diferentes áreas entre dos o más deportistas y brinda la posibilidad de exportar todos los resultados a formato Excel. (Palomeque, 2006)

X-Training Fussion: Es una herramienta aplicable a entrenadores personales, preparadores físicos, técnicos de diferentes deportes, clubes, instituciones deportivas, laboratorios de evaluaciones deportivas. Se pueden realizar planes de entrenamiento con gran rapidez, flexibilidad y eficiencia, gracias a una gran variedad de información y herramientas disponibles para poder utilizarlas de inmediato en cada una de las sesiones de entrenamiento. También da la posibilidad de confeccionar planificaciones para un deportista individual, o bien para un equipo de cualquier deporte. Puede almacenar simultáneamente infinita cantidad de planificaciones, para uno o más equipos o individuos y para todas las categorías que desee. Posee un generador de evaluaciones diseñado para que se pueda definir cualquier tipo de evaluaciones. Además, puede calcular resultados adicionales utilizando fórmulas que combinen todos los resultados obtenidos. Para ahorrar tiempo y esfuerzo, este sistema utiliza una herramienta que permite exportar las planillas de evaluaciones a Excel. Incorpora en forma automática el traspaso de información hacia la base de datos para poder analizar los resultados y obtener todo tipo de comparaciones y datos estadísticos (Palomeque, 2006). Toda esta gestión de datos resulta importante fuente de consulta y guía para la confección del sistema que se propone.

Bodymetrix: Es un software destinado a antropometrías, que propone trabajar de un modo más eficaz, ahorrando tiempo y esfuerzo. A la vez, permite ganar calidad y personalización en las presentaciones además de ser muy potente a la hora de entregar resultados e informes automáticos. Contiene una base de datos para almacenar todos los datos personales de los pacientes. Su diseño está desarrollado para la carga fácil y rápida de mediciones a través de una planilla de cálculos que se entrega con el programa, para luego incorporar automáticamente las mediciones y proceder con el cálculo correspondiente. Este proceso reduce el tiempo de procesamiento y facilita la tarea del profesional. Una vez calculados los resultados los esboza en gráficos a los cuales se puede personalizar los colores. Permite generar un resumen en un formato compatible con Microsoft Word® y puede modificarlo de acuerdo a sus necesidades. También incluye una historia clínica configurable, que está preestablecida dentro del software y que se puede personalizar y ajustar a las necesidades del usuario. (Palomeque, 2006)

1.3.2 Sistemas similares a nivel nacional

Hay que destacar que en Cuba no existe registro de algún software o aplicación destinado a la gestión de las pruebas médicas en el entrenamiento deportivo. Los sistemas que se documentan solo están dirigidos al entrenamiento deportivo y no a la gestión de pruebas médicas dentro de dicho entrenamiento, tal es el caso de los sistemas de planificación del entrenamiento en las pesas, en el atletismo y la esgrima que están desarrollados en el asistente de bases de datos Microsoft Access por la Facultad de Cultura Física de Villa Clara.

Recientemente se desarrolló el Sistema de Planificación del Entrenamiento del Judo Femenino que engloba todo el proceso de planificación del entrenamiento de las judocas en la Escuela Cubana de Judo Femenino. Este sistema está compuesto por cuatro módulos. El módulo administración es donde se gestionan los usuarios. El módulo plan gráfico permite realizar la planificación de las fechas de las pruebas médicas, competencias, campos de entrenamiento y test pedagógicos. Además, estará dividido por macrosistemas, mesosistemas y microsistemas para una mayor organización. Aunque gestiona la parte esencial del proceso de entrenamiento y planifica las fechas de realización de las pruebas médicas, este no se adentra en el control y gestión de las pruebas médicas dentro del ciclo de entrenamiento.

1.3.3 Resultados del estudio de los sistemas similares analizados

Luego de realizar una investigación sobre los diferentes sistemas utilizados en el ámbito nacional e

internacional para la planificación del entrenamiento de los diferentes deportes cabe destacar que no profundizan en las particularidades de las pruebas médicas aplicadas a los deportistas y no se adaptan a las necesidades de los médicos en la Escuela Cubana de Judo Femenino. A nivel mundial están los sistemas *Bodymetrix* y *X-Medalist* que se centran en el entrenamiento de los atletas y enfocan superficialmente el tema de la gestión de pruebas médicas con la confección de una historia médica de los deportistas y el sistema *X-Training Fusion* que no refiere poseer este historial pero posee herramientas internas de gestión de gráficos, evaluación de resultados y exportación de datos que sirven de ayuda y consulta para la implementación del sistema que se propone.

El estudio de sistemas nacionales arrojó que en nuestro país no existe un sistema que informatice la planificación de las pruebas médicas en el entrenamiento deportivo, señalando sus objetivos al entrenamiento de los atletas.

Es por esto que teniendo como punto de partida la investigación antes realizada y a propuesta de la Escuela Cubana de Judo Femenino se va a desarrollar un software que gestione las pruebas médicas aplicadas dentro de los períodos de entrenamiento de las judocas, así como la gestión de los resultados obtenidos en las mismas mediante reportes, gráficos y tablas que le faciliten el trabajo a los entrenadores, médicos y equipo técnico que utilizan esta información con el fin de mantener en óptimas condiciones la salud de las atletas.

1.4 Estudio de las metodologías y estándares para el desarrollo de software

1.4.1 Metodologías de desarrollo de software

Todo proceso de desarrollo de software para que se realice de manera exitosa debe estar basado en una metodología, que no es más que un conjunto de actividades con un orden lógico, en las cuales para cumplirlas se utilizan herramientas, técnicas y modelos, permitiendo obtener un software con calidad al concluir el proceso. No existe un proceso de desarrollo universal aplicable a todo proyecto. Factores como las características del equipo de desarrollo, el dominio de aplicación, el tipo de contrato, la complejidad y envergadura del proyecto, hacen necesario adaptar las metodologías de desarrollo. Las metodologías tradicionales presentan inconvenientes para realizar dicha adaptación, especialmente cuando se trata de

proyectos pequeños. Así, las metodologías ágiles han emergido como una interesante alternativa en este contexto considerándose un paradigma de desarrollo de software basado en procesos ágiles.

Una metodología debe ser lo suficientemente adaptable como para poder aplicarse en distintos proyectos, y lo suficientemente sencilla para que no resulte muy incómoda su utilización, pero a la vez suficientemente completa como para que su uso por parte de un equipo de desarrollo sea provechoso.

1.4.1.1 Programación Extrema (*Extreme Programming, XP*)

XP es una metodología ligera utilizada para proyectos de corto plazo y equipos pequeños. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Se define como especialmente adecuada para proyectos con funcionalidades imprecisas y muy cambiantes. Mientras que el Proceso Unificado de Desarrollo de Software (*Rational Unified Process, RUP*) intenta reducir la complejidad del software por medio de estructura y la preparación de las tareas pendientes en función de los objetivos de la fase y actividad actual, XP, como toda metodología ágil, lo intenta por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción. (Molperces, 2002)

La base para el desarrollo del software que usa esta metodología son las llamadas historia de usuarios, historias escritas por el cliente en las que describe escenarios sobre el funcionamiento del sistema y que no sólo están limitados a la interfaz de usuario, sino que también pueden describir modelos o dominios. Estas historias de usuarios junto a la arquitectura que se persigue, sirve de base para crear un plan de entregas de software entre el equipo de desarrollo y el cliente, para cada una de las cuales se definen los objetivos y las iteraciones (generalmente cortas) necesarias para cumplirlos. Las historias de usuarios y los casos de pruebas son la base sobre la que se asienta el trabajo del desarrollador. (Beck, 1999)

XP impone un alto nivel de disciplina entre los programadores. Permite mantener un mínimo nivel de documentación, lo cual a su vez se traduce en una gran velocidad en el desarrollo. (Schenone, 2004).

A continuación se presenta las características esenciales de XP organizadas en los tres apartados siguientes: historias de usuario, roles y proceso.

La Historia de los Usuarios

Las historias de usuario son la técnica utilizada en XP para especificar las necesidades del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales.

El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento las historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. (Letelier, 2008)

Respecto a la información contenida en la historia de usuario, existen varias plantillas sugeridas pero no existe un consenso al respecto. En muchos casos sólo se propone utilizar un nombre y una descripción. Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración.

Roles de XP

Para una mejor organización de la ejecución de las actividades para el desarrollo de proyectos guiados por esta metodología se establecen los siguientes roles: Programador, Cliente, Encargado de pruebas (*Tester*), Encargado de seguimiento (*Tracker*), Entrenador (*Coach*), Consultor, Gestor (*Big boss*).

Proceso XP

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1. (Letelier, 2008)

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. Se debe presionar al programador a realizar más trabajo que el estimado, ya que, de no ser así, perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste de seis fases:

1. Exploración.
2. Planificación de entrega (*Release*).
3. Iteraciones.
4. Producción.
5. Mantenimiento.
6. Muerte del Proyecto. (Letelier, 2008)

1.4.1.2 Proceso Unificado de Desarrollo de Software (RUP)

Es una metodología de desarrollo pesada para la ingeniería de software que va más allá del análisis y diseño orientado a objetos para proporcionar una familia de técnicas que soportan el ciclo completo de desarrollo de software. Unifica completamente a un equipo de desarrollo y optimiza la productividad de cada uno de los miembros del mismo, ayudando a los líderes de proyecto a incrementar su experiencia en el desarrollo de proyectos.

RUP se caracteriza por ser iterativo e incremental donde cada fase se desarrolla en iteraciones, de forma tal que se pueda dividir en pequeños proyectos mejorando su comprensión y desarrollo. También está dirigido por los casos de uso que no son más que el instrumento para validar la arquitectura del software y extraer los casos de prueba. Esta metodología se centra en la arquitectura, aquí los modelos son proyecciones del análisis y el diseño constituye la arquitectura del producto a desarrollar.

Como RUP es un proceso, en su modelación define como sus principales elementos:

- Trabajadores (quién).
- Artefactos (qué).

- Actividades (cómo).
- Flujo de actividades (cuándo).

Un proyecto que se desarrolle utilizando la metodología RUP está dividido en 4 fases, 6 flujos de ingeniería y 3 de apoyo.

Fases:

- Inicio.
- Elaboración.
- Construcción.
- Transición.

Flujos de ingeniería:

- Modelo de negocio.
- Requisitos.
- Análisis y diseño.
- Implementación.
- Prueba.
- Despliegue.

Flujos de apoyo:

- Gestión de cambios y configuraciones.
- Gestión de proyectos.
- Ambiente o entorno. (Kruchten, 2000)

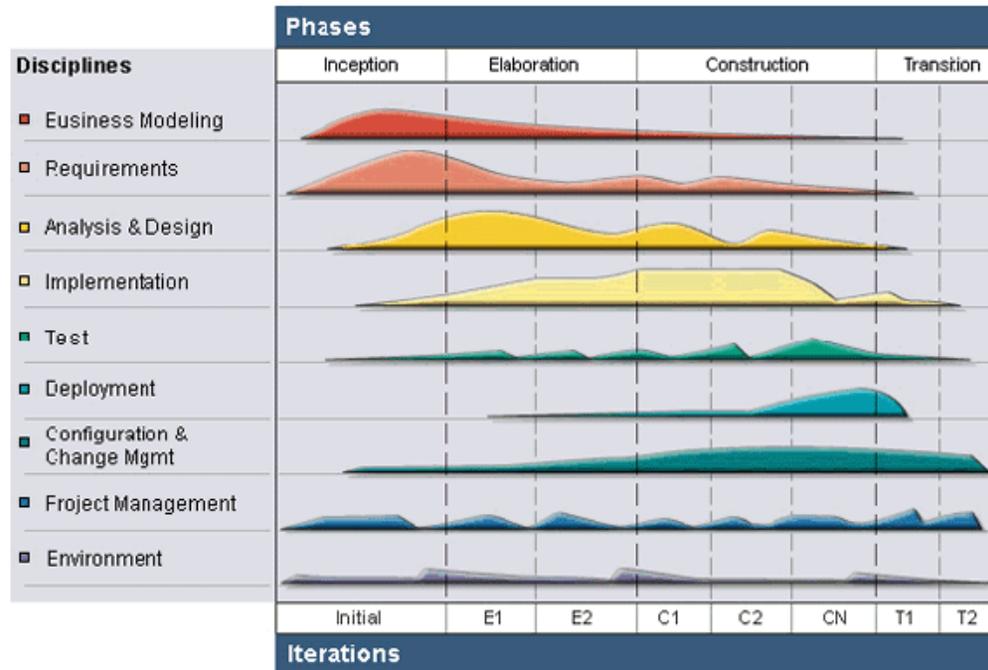


Figura 1: Fases y flujos de trabajo de RUP.

1.4.1.3 Desarrollo Guiado por la Funcionalidad (*Feature Driven Development, FDD*)

FDD es una metodología ágil orientada a la funcionalidad del software por sobre las tareas, sobre las cuales da guías mínimas. No hace énfasis en la obtención de los requerimientos sino en cómo se realizan las fases de diseño y construcción partiendo de una lista de características que debe reunir el software; provee estrategias de planeamiento para el líder de proyecto; fomenta la colaboración mediante la creación de equipos dirigidos por un programador jefe.

Un proyecto que sigue FDD se divide en 5 fases:

- Desarrollo de un modelo general.
- Construcción de la lista de funcionalidades.
- Plan de emisiones en base a las funcionalidades a implementar.
- Diseñar en base a las funcionalidades.
- Implementar en base a las funcionalidades. (Molperces, 2002)

Selección de la Metodología

Después de un estudio de algunas de las metodologías de desarrollo de software existentes, se llega a la conclusión de que todas, en gran medida, presentan cualidades bien destacables como es el caso de la sencillez, tanto en el aprendizaje como en la aplicación a un proyecto de software determinado, permitiendo de esta forma reducir los costos de implantación en un equipo de desarrollo. Sin embargo, se decidió escoger la metodología XP ya que propone que la comunicación y la satisfacción del cliente son lo principal y no se hace mucho énfasis en la documentación, es más importante definir las necesidades del cliente y las pruebas de calidad. XP es una metodología ligera utilizada para proyectos cuyo plazo de entrega sea corto.

Además, uno de los objetivos de importancia que aplica esta metodología es potenciar al máximo el trabajo en grupo, donde los jefes de proyecto, los clientes y desarrolladores son parte del equipo y están involucrados en el desarrollo del software. También es importante destacar que dada la poca experiencia que tiene el cliente en cuanto a lo que sitios Web respecta, las necesidades actuales pueden estar sujetas a futuros cambios y éste es un punto en el cual esta metodología es flexible ya que permite administrar estos cambios de forma óptima.

1.4.2 Lenguaje Unificado de Modelado (UML)

UML es un lenguaje de modelado para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos (OO). Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de software. Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que UML es un lenguaje de modelado, cuenta con reglas para combinar tales elementos. Ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. (Lafuente,1998)

Es importante destacar que UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso, sino un lenguaje que permite el modelado de sistemas con tecnología orientada a objetos. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se

les conoce como modelo. Además, describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

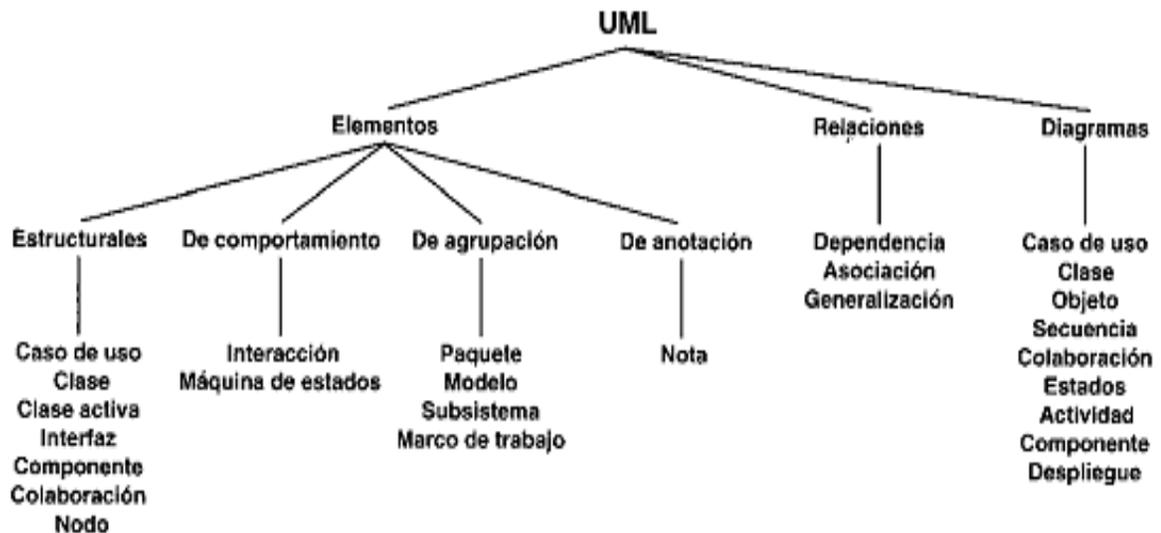


Figura 2: El vocabulario de UML.

1.5 Tendencias y tecnologías actuales. Selección de las herramientas y lenguajes de desarrollo.

Se propone utilizar tecnología Web para resolver el problema argumentado anteriormente, debido a que esta tecnología posee muchas ventajas como:

- Se necesita solamente un navegador Web y conexión al sistema.
- Se pueden ejecutar varios clientes simultáneamente y en diferentes estaciones de trabajo.
- Se necesitan pocos recursos para que la aplicación trabaje correctamente.
- Toda la información se encontrará de forma centralizada en el servidor, al igual que la administración.
- Por lo general no depende de ningún software.

- No hay generalmente discriminación del sistema operativo por parte del usuario, aunque exista por parte del servidor.

1.5.1 Lenguajes de programación

Conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores. El sistema que se implementará será una aplicación Web, por tanto, usará arquitectura cliente – servidor. En este caso el cliente solo estará relacionado con la interfaz externa del servidor, no con su lógica interna. Además, el cliente puede trabajar sin importar donde se encuentre el servidor, ni qué sistema operativo tenga, solamente necesita conexión hacia él. El servidor puede estar sujeto a cambios que deben influir poco o en nada en el cliente, también puede estar en diferente plataforma a la del servidor que no afectará el funcionamiento del sistema. Algo importante es que el servidor puede brindar servicios múltiples y simultáneos a clientes en cualquier lugar. Para llevar a cabo la implementación del sistema es necesario saber cual lenguaje de programación usar. Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de un ordenador. (Manzanedo, 1999)

Para desarrollar un sistema Web los lenguajes de programación se pueden dividir en:

- Lenguaje del lado del cliente.
- Lenguaje del lado del servidor.

Dentro de este último caso (lenguaje del lado del servidor), se pueden apreciar lenguajes como PHP, ASP, JSP, Java. Dentro de los lenguajes del lado del cliente se encuentran HTML, CSS, *JavaScript*.

Hypertext Pre – Processor (PHP)

Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web (Hinojoza, 2005), se usa para el desarrollo de contenidos de aplicaciones Web. Principalmente es usado en la interpretación del lado del servidor (*server-side scripting*), pero puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica.

Características:

- Es un lenguaje multiplataforma.
- Tiene capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- Posee la capacidad de expandir su potencial utilizando una enorme cantidad de módulos (llamados extensiones).
- Posee una amplia documentación entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos (POO).
- Posee una biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde PHP5).

Active Server Pages (ASP)

Es una tecnología del lado del servidor desarrollada por *Microsoft* para la realización de sitios Web dinámicos. Las páginas Web desarrolladas bajo este lenguaje necesitan tener instalado *Internet Information Server (IIS)*.

ASP no necesita ser compilado para ejecutarse. Existen varios lenguajes que se pueden utilizar para crear páginas ASP. El más utilizado es *VBScript*, nativo de *Microsoft*. ASP se puede hacer también en Perl y *JavaScript*. El código ASP puede ser insertado junto con el código HTML. Los archivos cuentan con la extensión (asp).

Características

- Usa *Visual Basic Script*, siendo fácil para los usuarios.
- Comunicación óptima con SQL Server.
- Soporta el lenguaje *JavaScript* de *Microsoft*.
- Se necesita escribir mucho código para realizar funciones sencillas.

- Es una tecnología propietaria.

Java Server Pages (JSP)

Es un lenguaje multiplataforma para la creación de sitios Web dinámicos. Está orientado a desarrollar páginas Web en Java y creado para ejecutarse del lado del servidor.

JSP fue desarrollado por *Sun Microsystems*. Comparte ventajas similares a las de ASP.NET desarrollado para la creación de aplicaciones Web potentes. Posee un motor de páginas basado en los *servlets* de Java. Para su funcionamiento se necesita tener instalado un servidor *Tomcat*. (Arambillette, 2003)

Características:

- El código se encuentra separado de la lógica del programa.
- Las páginas son compiladas en la primera petición.
- Permite separar la parte dinámica de la estática en las páginas Web.
- Los archivos se encuentran con la extensión (jsp).
- El código JSP puede ser incrustado en código HTML.
- Tiene alta complejidad de aprendizaje.

Java

Es un lenguaje de programación orientado a objetos. Tiene un modelo de objetos simple y elimina herramientas de bajo nivel que suelen inducir a muchos errores como la manipulación directa de punteros o memoria.

Características

- Orientado a Objetos: método de programación y diseño del lenguaje. Aunque hay muchas interpretaciones, una primera idea es diseñar el software de forma que los distintos tipos de datos que se usen estén unidos a sus operaciones. (Alvarez, 2003)
- Independencia de la plataforma: significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Es lo que significa ser capaz de escribir un

programa una vez y que pueda ejecutarse en cualquier dispositivo, tal como reza el axioma de Java, “*write once, run everywhere*”.

- Posee también recolector automático de basura lo cual independiza al programador de tener que administrar memoria solicitada dinámicamente de forma manual.

JavaScript

JavaScript es muy fácil de aprender para quien ya conoce lenguajes similares como el C++ o Java, pero, dada su simplicidad sintáctica y su manejabilidad, no es tampoco difícil para quien se acerca por primera vez a este lenguaje. Sin embargo, esto puede ser un arma de doble filo porque la simplicidad se basa en una disponibilidad de objetos limitada, por lo que algunos procedimientos, aparentemente muy sencillos, requieren *scripts* bastante complejos (Valdelli, 2006).

JavaScript es un lenguaje de programación interpretado, es decir, que no requiere compilación, ya que el lenguaje funciona del lado del cliente y los navegadores son los encargados de interpretar estos códigos. Además, *JavaScript* tiene la ventaja de ser incorporado en cualquier página Web, puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado. Es un lenguaje interpretado basado en objetos. No es un lenguaje de sintaxis estricta en lo que a tipos de datos se refiere: las variables no necesariamente deben ser declaradas. La verificación de objetos se lleva a cabo en tiempo de corrida. Es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox.

Lenguaje de Mercado de Hipertexto (HyperText Markup Language, HTML)

HTML es el lenguaje de marcas de texto utilizado normalmente en la WWW (*World Wide Web*). Fue creado en 1986 por el físico nuclear Tim Berners-Lee, el cual tomó dos herramientas preexistentes: El concepto de hipertexto (conocido también como *link* o ancla) el cual permite conectar dos elementos entre sí y el SGML (Lenguaje Estándar de Marcación General) el cual se utiliza para colocar etiquetas o marcas en un texto que indique como debe verse. (Lafuente, 1998)

HTML no es propiamente un lenguaje de programación como C++ o *Visual Basic*, sino un sistema de etiquetas. No presenta ningún compilador, por lo tanto, algún error de sintaxis que se presente éste no lo detectará y se visualizará en la forma como éste lo entienda. Los documentos pueden ser mostrados por los visores o "*browsers*" de páginas Web en Internet, como *Netscape Navigator*, *Mosaic*, *Opera* y *Microsoft*

Internet Explorer. También existe el HTML dinámico (DHTML), que es una mejora de *Microsoft* de la versión 4.0 de HTML que le permite crear efectos especiales como por ejemplo los efectos de transición al estilo de anuncio publicitario giratorio entre página y página.

Hojas de Estilo en Cascada (*Cascading Style Sheets*, CSS)

Es la tecnología desarrollada por el *World Wide Web Consortium* (W3C) con el fin de separar la estructura de la presentación. A pesar de que la recomendación oficial del grupo de trabajo de la W3C ya había alcanzado la estabilidad requerida para que fuera soportada por los principales navegadores comerciales, como *Netscape* e *Internet Explorer*. (Álvarez, 2000)

La potencia de la tecnología CSS permite aplicar al documento formato de modo mucho más exacto. Si antes el HTML se quedaba corto para maquetar las páginas y había que utilizar trucos para conseguir los efectos deseados, ahora existen muchas más herramientas que permiten definir esta forma:

- Se puede definir la distancia entre líneas del documento.
- Colocar elementos en la página con mayor precisión y sin dar lugar a errores.
- Definir la visibilidad de los elementos, márgenes, subrayados y tachados.

Esta tecnología es bastante nueva, por lo que no todos los navegadores la soportan. En concreto, sólo los navegadores de *Netscape* versiones de la cuatro en adelante y de *Microsoft* a partir de la versión tres son capaces de comprender los estilos en sintaxis CSS.

Selección del lenguaje de desarrollo.

Después de haber realizado un estudio, se llega a la conclusión de que Java es un lenguaje para aplicaciones con alta complejidad y seguridad, además una de sus desventajas es que el tiempo en que se accede a las páginas, es lento, en este caso, la solución implica mejorar el equipo con el que se cuenta (modem, RAM, caché). ASP es una tecnología propietaria por lo que se decide descartarla, además de que se necesita mucho código para hacer funciones sencillas. En cuanto a JSP es un lenguaje para hacer Web potentes, su entendimiento es complejo y para su funcionamiento se necesita de otras herramientas. Por su parte PHP es el lenguaje adecuado para desarrollar la aplicación del lado del servidor, ya que es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor Web en forma de módulo o ejecutado como un binario, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red

en el servidor. La aplicación que se va a desarrollar no es grande y PHP es un lenguaje adecuado para pequeñas aplicaciones Web dinámicas. También es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo Web y puede ser incluido dentro del código HTML.

Del lado del cliente HTML y CSS son los indicados pues se podrá manejar con más facilidad todo lo referente al diseño, ya que HTML indica al navegador donde colocar cada texto, imagen o video y la forma que tendrán estos al ser colocados en la página. En cuanto a CSS se podrá incluir márgenes, tipos de letra, fondos, colores, incluso definir los estilos en un archivo externo a nuestras páginas; así, si en algún momento se quiere cambiar alguno de ellos, automáticamente se actualizarán todas las páginas vinculadas del sitio. Para la validación de los datos que se introducen al sistema se decide la utilización de *Javascript* para hacer la Web más interactiva y validar la entrada de los datos en el sistema. Por sus ventajas en cuanto a ser incorporado en cualquier página Web, puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado y es un lenguaje interpretado basado en objetos.

1.5.2 Entorno de Desarrollo Integrado (IDE).

Un IDE es un entorno de programación que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Estas pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

Zend Studio For Eclipse

Este IDE está diseñado para desarrolladores profesionales de PHP, combina un IDE versátil y potente con las capacidades de expansión del ecosistema del proyecto Eclipse. Entre sus funcionalidades se destaca la capacidad de refactorización del código fuente, funcionalidad que permite adecuar el comportamiento externo de una función o clase sin cambiar el funcionamiento interno que junto a los nuevos asistentes y capacidades de generación de código facilitan el trabajo a los desarrolladores. Desde el punto de vista de un IDE completo, disponer de un buen depurador local con la conexión a los servidores de desarrollo, junto a una política de trabajo en equipo y un sistema de control de versiones es posible manejar sin problemas proyectos complejos en PHP.

Eclipse

Eclipse es una plataforma de desarrollo de código abierto basada en Java. Es un desarrollo de IBM cuyo código fuente fue puesto a disposición de los usuarios. En sí mismo Eclipse es un marco y un conjunto de servicios para construir un entorno de desarrollo a partir de componentes conectados (*plug-in*). (Gutiérrez, 2004).

Hay *plug-ins* para el desarrollo de Java (*JDT Java Development Tools*) así como para el desarrollo en C/C++, COBOL, PHP.

Este IDE demuestra su calidad en el desarrollo de PHP con el *plug-in* para este lenguaje. Este *plug-in* además permite que el trabajo con PHP sea muy rápido pues tiene auto completamiento y colorea errores.

Es únicamente un armazón sobre el que se pueden montar herramientas de desarrollo para cualquier lenguaje, mediante la implementación de los *plug-ins* adecuados. La arquitectura de Eclipse permite, además de integrar diversos lenguajes sobre un mismo IDE, introducir otras aplicaciones accesorias que pueden resultar útiles durante el proceso de desarrollo como: herramientas UML, editores visuales de interfaces y ayuda en línea para librerías.

Selección del IDE de desarrollo.

El Eclipse es el IDE seleccionado para desarrollar la propuesta ya que posee entre otras características que es libre. Este es un buen entorno de programación que puede competir con cualquier paquete de software comercial. El *Zend Studio For Eclipse* es un entorno muy bueno para el desarrollo de aplicaciones Web, pero requiere licencia de pago, no incluye editor visual HTML y es un poco complejo.

1.5.3 Herramientas CASE de Modelado con UML.

Las herramientas CASE (*Computer Aided Software Engineering*: Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y costo.

Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores.

Visual Paradigm

Es una herramienta CASE que utiliza UML como lenguaje de modelado profesional la cual soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite realizar ingeniería tanto directa como inversa. La herramienta es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto. Genera la documentación del proyecto automáticamente en varios formatos como Web o PDF y permite control de versiones. Cabe destacar igualmente su robustez, usabilidad y portabilidad.

Características

- Usa un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Modelo y código permanecen sincronizados en todo el ciclo de desarrollo.
- Posee disponibilidad en múltiples plataformas.
- Es amigable.
- Permite la generación de documentos.
- Posee interoperabilidad con otras aplicaciones.
- Permite la integración con distintos Ambientes de Desarrollo Integrados (IDE).
- Tiene facilidad de instalación y actualización. (Hernandis, 2005)

Rational Rose

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML 1.1. Propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando un modelo completo que representa el dominio del problema y el sistema de software. (Taboada, 2005)

Desarrollo Iterativo: utiliza un proceso de desarrollo iterativo controlado donde el desarrollo se lleva a cabo en una secuencia de iteraciones.

Trabajo en Grupo: permite que haya varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.

Ingeniería Inversa: a partir del código de un programa, se puede obtener información sobre su diseño.

Selección de la herramienta case para UML.

Por las características vistas anteriormente se decide seleccionar *Visual Paradigm*, ya que se integra mejor por su rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Posee disponibilidad de múltiples plataformas y la Universidad de las Ciencias Informáticas paga su licencia de manera indefinida por lo que puede utilizarse libremente para el desarrollo de software.

1.5.4 Sistema Gestor de Bases de Datos (SGBD)

Un Sistema Gestor de Bases de Datos (SGBD) o DBMS (*DataBase Management System*) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos garantizando la seguridad e integridad de los mismos.

PostgreSQL

Es un gestor de base de datos que funciona en los principales sistemas operativos, dentro de ellos Linux. Posee una documentación muy bien organizada, cuenta con comunidades muy activas y con bajo costo de propiedad total. Es un gestor altamente adaptable a las necesidades del cliente. Brinda soporte nativo para los lenguajes de programación más populares, dentro de ellos PHP. Posee utilidades para la limpieza de las bases de datos (*Vacuum*) y también varios tipos de índices aplicativos de comparación de bases de datos para testear replicaciones. Algunas bondades que ofrece este gestor son:

Gran escalabilidad: Es capaz de ajustarse al número de procesadores y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones

simultáneas de manera correcta (en algunos puntos de referencia se dice que ha llegado a soportar el triple de carga de lo que soporta Misal). (ALIAGA, 2008)

Implementa el uso de reversiones, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz y ofreciendo soluciones en campos en las que Misal no podría.

Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel como puede ser Oracle. (ALIAGA, 2008)

Misal

Es la base de datos de código abierto más popular a nivel mundial. Su continuo desarrollo y su creciente popularidad están haciendo de Misal un competidor cada vez más directo de gigantes en la materia de las bases de datos como Oracle (Manuel, 2005). Misal es un DBMS para bases de datos relacionales.

Existen varios tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. Como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información, fue escrito en C y C++ y se destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos. (Manuel, 2005) Entre sus principales características se encuentran:

- Interioridades y portabilidad.
- Escrito en C y en C++.
- Probado con un amplio rango de compiladores diferentes.
- Funciona en diferentes plataformas. Usa *GNU Autopase*, *Autocine* y *Libor* para portabilidad.
- Uso completo de multi – hilos mediante hilos del *kernel* Pueden usarse fácilmente múltiples procesadores si están disponibles.
- Proporciona sistemas de almacenamiento transaccional y no transaccional.
- Relativamente sencillo de añadir otro sistema de almacenamiento. Esto es útil si desea añadir una interfaz SQL para una base de datos propia.

- Uniones muy rápidas usando una multi – unión de un paso optimizado.
- Tablas *hash* en memoria, que son usadas como tablas temporales.

Oracle

Oracle es básicamente una herramienta cliente – servidor para la gestión de base de datos, es un producto vendido a nivel mundial aunque la gran potencia que tiene y su elevado precio hace que sólo se vea en empresas muy grandes y multinacionales. Como es un sistema muy caro no está tan extendido como otras bases de datos, por ejemplo, Access, MySQL y SQL Server. El software que produce Oracle no sólo soporta datos alfanuméricos ubicados en las tradicionales filas y columnas de las bases de datos, sino que también soporta textos sin estructura, imágenes, audio y video. Puede ser usado tanto para el manejo de información personal como para gigantescas bibliotecas multimedia. Corre en equipos desde la más pequeña laptop hasta la mayor supercomputadora.

Posee igual interacción en todas las plataformas (*Windows, Unix, Macintosh y Mainframe*). Soporta bases de datos de todos los tamaños, desde severas cantidades de bytes, gigabytes hasta terabytes en tamaño. Provee salvar con seguridad de error lo visto en el monitor y la información de acceso y uso. Soporta también un verdadero ambiente cliente – servidor. Además de establecer un proceso entre bases de datos del servidor y el cliente para la aplicación de programas.

Selección del Sistema Gestor de Bases de Datos a utilizar

Después de haber consultado la información de los SGBD antes expuestos se elige el *PostgreSQL* como el gestor que se utilizará en el desarrollo del sistema, ya que posee una gran escalabilidad, pues es capaz de ajustarse al número de procesadores y a la cantidad de memoria que posee el sistema de forma óptima, teniendo soporte para una mayor cantidad de peticiones simultáneas de manera correcta. También implementa el uso de reversiones, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz y ofreciendo soluciones en campos en las que MySQL no podría. Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel. Por su parte Oracle tiene la desventaja de ser privativo además de tener un elevado precio en el mercado, lo que hace que sólo se observe en empresas muy grandes o multinacionales por lo general, provocando que no esté extendido en el desarrollo de sitios Web.

1.6 Conclusiones

En este capítulo se obtuvo como resultado la premisa de que tanto a nivel nacional como internacional existen sistemas que abordan temas relacionados con la información médica de los deportistas pero no poseen módulos de gestión de pruebas médicas que profundicen en el proceso de planificación y control de los datos arrojados por estas pruebas. Se utilizarán las técnicas de recopilación de la información adecuadas como el uso del lenguaje unificado de modelado (UML), cuya utilización de diagramas y gráficos brinda una mejor perspectiva de lo que se quiere. *Visual Paradigm* es la herramienta de modelado a usar, ya se integra mejor que las otras herramientas estudiadas por su rápida construcción de aplicaciones de calidad, mejores y a un menor costo. PHP es el lenguaje a utilizar pues es poderoso para desarrollar aplicaciones Web del lado del servidor. El uso de *JavaScript* permite hacer la Web más interactiva y posibilita también que muchas validaciones se realicen del lado del cliente. El Eclipse es el IDE que se va a utilizar para desarrollar la propuesta al problema, ya que tiene como ventaja sobre los demás que es libre y tiene un *plug – in* para PHP que facilita el trabajo a la hora de programar en este lenguaje.

Capítulo 2: Desarrollo de la solución

2.1 Introducción

En el presente capítulo se describen brevemente las principales funciones y el flujo actual de los procesos involucrados en el problema en cuestión, haciéndose un análisis de cómo se ejecuta actualmente el proceso de gestión de las pruebas médicas llevado a cabo en la Escuela Cubana de Judo Femenino. Se detallan los artefactos de la metodología que se propone en el capítulo anterior, XP, enmarcando el trabajo en sus tres primeras fases: exploración, planificación e iteraciones.

Se muestran las historias de usuarios que fueron escritas por el cliente y se definen tres iteraciones por los programadores, elaborándose el Plan de Entrega para cada versión.

2.2 Descripción de las acciones vinculadas al campo de acción

Actualmente la gestión de las pruebas médicas en la Escuela Cubana de Judo Femenino se hace muy engorrosa, esto se debe a que es necesaria la ejecución de extensos cálculos con fórmulas de alta complejidad, así como la obtención y utilización de datos dispersos en diferentes tablas, como escalas y mediciones que traen consigo errores en los resultados obtenidos. La información arrojada durante el proceso de aplicación de las pruebas médicas no se encuentra centralizada, tampoco se cuenta con un mecanismo para gestionar los resultados alcanzados en la aplicación de las pruebas médicas a las judocas en sus ciclos de entrenamiento.

Lo antes planteado constituye un inconveniente para la obtención de resultados fiables y para la rapidez del proceso de gestión de la información de diagnóstico de las condiciones competitivas de las atletas.

2.2.1 Flujo actual de eventos

El proceso desarrollado por los médicos como parte de sus acciones para el diagnóstico de las atletas en la actualidad está definido por una serie de actividades que fluye de la siguiente forma:

Durante la planificación del entrenamiento deportivo de las judocas con vista a ciclos competitivos se tiene en cuenta la aplicación del test pedagógico. Este momento es el que toman los médicos como punto de partida para planificar las pruebas médicas que se les realizarán a las atletas para su diagnóstico. Dentro

de esta planificación se definen un gran número de exámenes que comprenden las pruebas de terreno (1500 metros y test de proyecciones), pruebas de laboratorio (clínico, cardiovascular y desarrollo físico) y pruebas de urea.

En este proceso de obtención de los resultados de estos exámenes es necesaria la utilización de mediciones, reglajes y escalas que se encuentran dispersas y no están digitalizadas, esto trae como resultado el deterioro debido al uso y extensión de los ciclos de entrenamiento. Los datos obtenidos de las mismas se utilizan para calcular valores de indicadores mediante fórmulas de alta complejidad que introducen errores y dificultan el trabajo de los médicos.

Para un mejor diagnóstico de las judocas los médicos realizan un resumen de los resultados obtenidos en un ciclo de aplicación de pruebas médicas, este proceso ocurre de forma manual por lo que pueden perderse y omitirse datos importantes. De esta forma se obstaculiza la idea de que puedan ser utilizados como material de consulta en ciclos posteriores.

Por otra parte, no poseen un instrumento que integre los resultados de las condiciones de salud de las judocas en todo este ciclo de entrenamiento que puede tomar meses.

2.2.2 Objetos de automatización

Durante el ciclo de desarrollo de la planificación y gestión de la información de las pruebas médicas llevadas a cabo en la Escuela Cubana de Judo Femenino existen varias acciones que deben ser automatizadas, puesto que se hace engorroso el manejo de los datos y se tiende a cometer a errores.

Se automatizarán las actividades de pesaje diario de las judocas, la realización de los cálculos necesarios para obtener los resultados de las pruebas de terreno, de laboratorio así como el registro de los resultados de las pruebas de urea. Se elaborará una historia clínica donde se recojan los datos de las condiciones de salud de las judocas durante todo el período de entrenamiento.

2.2.3 Propuesta del sistema

Con la realización del siguiente trabajo se dispone desarrollar la implementación del módulo de pruebas médicas del sistema de planificación de entrenamiento del judo femenino que permita la gestión (insertar, modificar y consultar) del conjunto de información referente a las pruebas médicas aplicadas a las judocas de la Escuela Cubana de Judo Femenino, dentro de ellas, pruebas de terreno, de laboratorio así como el

pesaje diario de las atletas y el registro de los resultados de las pruebas de urea. Se realizarán los cálculos necesarios para la evaluación y obtención de los indicadores analizados por los médicos para el posterior diagnóstico de las atletas.

Existirá un espacio dedicado a la publicación de las historias clínicas de todas las judocas donde se recogerán todos los padecimientos, enfermedades y lesiones de cada una de ellas. Se habilitará un solo usuario, médico, para limitar el acceso a los servicios brindados por dicho módulo a usuarios que no lo requieran.

2.3 Desarrollo de la solución

2.3.1 Fases propuestas por la metodología

Fase I: Exploración

Como la metodología XP propone, en esta fase los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo, el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. (Penadés, 2008)

Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La extensión de la fase de exploración se sugiere que sea de 2 ó 3 semanas hasta pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

Para el desarrollo del módulo que se desea implementar solo se utilizarán dos semanas, debido a que el mismo no es un sistema de gran extensión y el equipo de desarrollo tiene dominio del trabajo con las herramientas que se utilizarán.

Fase II: Planificación y entrega

La planificación es una fase corta en la que el cliente, los gerentes y el grupo de desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario y asociadas a éstas, las entregas. Típicamente esta fase consiste en una o varias reuniones grupales de planificación. El resultado de esta fase es un Plan de Entregas, o "*Release Plan*". (Joskowicz, 2008)

En esta fase el cliente establece la prioridad de cada historia de usuario y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debe obtenerse en no más de tres meses.

Fase III: Iteraciones

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una, un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo. Las iteraciones son también utilizadas para medir el progreso del proyecto.

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto de iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fuercen la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor del negocio). Al final de la última iteración el sistema estará listo para entrar en producción. (Penades, 2008)

2.3.2 Personas relacionadas con el sistema

En la siguiente tabla (Tabla 1) se definen las personas que de una forma u otra van a interactuar con la aplicación, incluyendo los que mantienen el sistema actualizado, en correcto funcionamiento y los que utilizan sus servicios.

Tabla 1. Personas relacionadas con el sistema.

Personas relacionadas con el sistema	Descripción
Médico	Es la persona que utiliza el sistema para la gestión de

	la planificación y el registro de los resultados médicos obtenidos luego de la aplicación de las pruebas médicas a las judocas durante el ciclo de entrenamiento. Tiene la posibilidad de acceder a todos los contenidos y servicios que estén disponibles para su rol en el sistema.
Administrador	Es la persona que realiza la gestión de todo el contenido del sistema. Es el encargado de administrar las diferentes cuentas de los usuarios autenticados en la aplicación.

2.3.3 Historias de Usuario

En XP la gestión de requerimientos del sistema es extremadamente simple, el cliente describe y prioriza sus necesidades mediante historias de usuario. Son descripciones cortas y escritas sin terminología técnica.

Las historias de usuario solamente proporcionaran los detalles sobre la estimación del riesgo y cuánto tiempo conllevará la implementación de dicha historia de usuario. El nivel de detalle de las historias de usuario debe ser el mínimo posible que permita hacerse una ligera idea de cuánto costará implementar el sistema.

Los programadores estiman el esfuerzo asociado y las dependencias entre ellas. Para planificar el trabajo desde el punto de vista técnico, las historias de usuario son divididas en tareas para las cuales también se realiza una estimación. Teniendo en cuenta el esfuerzo asociado a las historias de usuario y las prioridades del cliente se define una versión que sea de valor y que tenga una duración de unos pocos meses.

Tabla 2. HU Plan gráfico médico.

Historia de Usuario	
Número: 1	Nombre Historia de Usuario: Plan gráfico médico

Modificación de Historia de Usuario número:	
Usuario: Médico	Iteración Asignada: 1
Prioridad de Negocio: Alta (Alta/ Media/ Baja)	Puntos Estimados: 0.6
Riesgo de Desarrollo: Baja (Alta/ Media/ Baja)	Puntos Reales: 0.4
Descripción: Se podrán consultar los datos de la planificación de las diferentes pruebas médicas que se desarrollarán dentro del entrenamiento de las judocas, además de exportar a documento PDF el mismo.	
Observaciones:	

Tabla 3. HU Peso de atletas.

Historia de Usuario	
Número: 2	Nombre Historia de Usuario: Peso de atletas
Modificación de Historia de Usuario número:	
Usuario: Médico	Iteración Asignada: 1
Prioridad de Negocio: Alta (Alta/ Media/ Baja)	Puntos Estimados: 0.8
Riesgo de Desarrollo: Baja (Alta/ Media/ Baja)	Puntos Reales: 0.8
Descripción: Se lleva un control diario del peso de las judocas para evaluar su forma física, se podrán insertar, modificar, consultar los datos en el sistema, además de exportar los datos a un documento PDF y graficar los valores de peso hasta la fecha. También se obtiene un reporte con los resultados del	

pesaje diario de cada atleta en fechas anteriores.

Observaciones:

Tabla 4. HU Prueba de 1500 metros.

Historia de Usuario	
Número: 3	Nombre Historia de Usuario: Prueba de 1500 metros
Modificación de Historia de Usuario número:	
Usuario: Médico	Iteración Asignada: 1
Prioridad de Negocio: Alta (Alta/ Media/ Baja)	Puntos Estimados: 0.8
Riesgo de Desarrollo: Baja (Alta/ Media/ Baja)	Puntos Reales: 0.8
<p>Descripción:</p> <p>Se introducen los resultados de las pruebas realizadas a las atletas en el entrenamiento y se calculan los datos necesarios para la evaluación médica de las judocas en dichos exámenes; podrán insertar, modificar, consultar los datos en el sistema, además de exportarlos a un documento PDF. También se obtiene un reporte con los resultados de cada atleta en pruebas anteriores.</p>	
Observaciones:	

Tabla 5. HU Test de proyecciones.

Historia de Usuario	
Número: 4	Nombre Historia de Usuario: Test de proyecciones
Modificación de Historia de Usuario número:	
Usuario: Médico	Iteración Asignada: 2

Prioridad de Negocio: Media (Alta/ Media/ Baja)	Puntos Estimados: 0.8
Riesgo de Desarrollo: Baja (Alta/ Media/ Baja)	Puntos Reales: 0.8
Descripción: Se introducen los resultados de los ejercicios de proyección efectuados por las atletas durante el entrenamiento y se calculan los datos necesarios para su evaluación médica durante el test de proyecciones; el médico podrá insertar, modificar, consultar los datos en el sistema, además de exportar los datos a un documento PDF. También se obtiene un reporte con los resultados de cada atleta en pruebas anteriores.	
Observaciones:	

Tabla 6. HU Laboratorio clínico.

Historia de Usuario	
Número: 5	Nombre Historia de Usuario: Laboratorio clínico
Modificación de Historia de Usuario número:	
Usuario: Médico	Iteración Asignada: 2
Prioridad de Negocio: Media (Alta/ Media/ Baja)	Puntos Estimados: 0.8
Riesgo de Desarrollo: Baja (Alta/ Media/ Baja)	Puntos Reales: 0.8
Descripción: Se introducen en el sistema los resultados de los análisis realizados a las judocas en el laboratorio clínico y se calculan los datos necesarios para su evaluación médica; podrán insertar, modificar,	

Capítulo 2: Desarrollo de la solución

consultar los datos y también exportar los mismos a un documento PDF. También se obtiene un reporte con los resultados de cada atleta en pruebas anteriores.

Observaciones:

Tabla 7. HU Laboratorio de desarrollo físico.

Historia de Usuario	
Número: 6	Nombre Historia de Usuario: Laboratorio de desarrollo físico
Modificación de Historia de Usuario número:	
Usuario: Médico	Iteración Asignada: 2
Prioridad de Negocio: Media (Alta/ Media/ Baja)	Puntos Estimados: 0.8
Riesgo de Desarrollo: Baja (Alta/ Media/ Baja)	Puntos Reales: 0.8
<p>Descripción:</p> <p>Se introducen los resultados de los análisis realizados en el laboratorio de desarrollo físico al sistema y se calculan los datos necesarios para la evaluación médica de las judocas. Se podrán insertar, modificar, consultar los datos en el sistema, así como exportarlos a un documento PDF. También se obtiene un reporte con los resultados de cada atleta en pruebas anteriores.</p>	
Observaciones:	

Tabla 8. HU Laboratorio cardiovascular.

Historia de Usuario	
Número: 7	Nombre Historia de Usuario: Laboratorio cardiovascular
Modificación de Historia de Usuario número:	

Capítulo 2: Desarrollo de la solución

Usuario: Médico	Iteración Asignada: 2
Prioridad de Negocio: Media (Alta/ Media/ Baja)	Puntos Estimados: 0.6
Riesgo de Desarrollo: Bajo (Alta/ Media/ Baja)	Puntos Reales: 0.6
Descripción: Se introducen los resultados arrojados durante el proceso de análisis de las condiciones cardiovasculares de las judocas en el sistema y se calculan los datos necesarios para su evaluación médica; podrán insertar, modificar, consultar los datos en el sistema y también exportarlos a un documento PDF. También se obtiene un reporte con los resultados de cada atleta en pruebas anteriores.	
Observaciones:	

Tabla 9. HU Historia clínica.

Historia de Usuario	
Número: 8	Nombre Historia de Usuario: Historia clínica
Modificación de Historia de Usuario número:	
Usuario: Médico	Iteración Asignada: 3
Prioridad de Negocio: Media (Alta/ Media/ Baja)	Puntos Estimados: 0.4
Riesgo de Desarrollo: Bajo (Alta/ Media/ Baja)	Puntos Reales: 0.4
Descripción: Se podrá obtener una historia clínica de cada una de las atletas con los padecimientos,	

enfermedades y lesiones. Se podrán insertar, modificar, consultar los datos en el sistema y exportarlos a un documento PDF.
Observaciones:

Tabla 10. HU Prueba de urea.

Historia de Usuario	
Número: 9	Nombre Historia de Usuario: Prueba de urea
Modificación de Historia de Usuario número:	
Usuario: Médico	Iteración Asignada: 3
Prioridad de Negocio: Media (Alta/ Media/ Baja)	Puntos Estimados: 0.6
Riesgo de Desarrollo: Baja (Alta/ Media/ Baja)	Puntos Reales: 0.6
Descripción: Se lleva un control de los resultados de las pruebas de urea que se realiza a las judocas, se podrán insertar, modificar, consultar los datos en el sistema, además de exportar los datos a un documento PDF. También se obtiene un reporte con los resultados de cada atleta en pruebas anteriores.	
Observaciones:	

2.3.4 Iteraciones

Iteración 1

Esta iteración tendrá como objetivo darle cumplimiento a las historias de usuario 1, 2 y 3 que representan un mayor valor para el cliente, pues con las mismas se conformará la base de la estructura del negocio.

Estas recogen las principales funcionalidades del sistema, pues a través de ellas se definen aspectos que serán utilizados luego por las demás funcionalidades. Se mostrará la planificación de la realización de las pruebas médicas por el personal médico a través del plan gráfico y se implementará la funcionalidad que gestiona el pesaje diario de las judocas. También se implementará todo lo referente a las pruebas de terreno (prueba 1500 metros) que posee gran importancia para el diagnóstico de la forma física de las judocas.

Iteración 2

Esta iteración está centrada en desarrollar una parte de los requerimientos con prioridad media en el negocio. Está conformada por las historias de usuario 4, 5, 6 y 7 que no poseen un nivel de riesgo muy elevado para el programador.

En ellas se gestionará la información referente a los resultados arrojados por los test de proyecciones, laboratorios clínicos, cardiovasculares y de desarrollo físico, enmarcados en el diagnóstico de las judocas en cuanto a pruebas de laboratorio.

Iteración 3

En esta iteración se da continuidad a los objetivos de la iteración anterior pues se enmarca en los restantes requerimientos definidos con prioridad media, por lo que abarca las historias de usuario 8 y 9. En esta iteración se desarrollará la creación de una historia clínica que contiene un resumen de los padecimientos, enfermedades y lesiones de las judocas. Se gestionarán además los datos arrojados por la prueba de urea realizada a cada una de las atletas y se brindará un reporte de los resultados obtenidos por cada atleta en la prueba seleccionada en períodos anteriores.

2.3.5 Estimación del esfuerzo por historia de usuario

Tabla 11. Estimación del esfuerzo por historia de usuarios.

Historias de usuario	Tiempo estimado (semana ideal de trabajo)	Iteración asignada	Tiempo real
Plan Gráfico Médico	2	1	2

Peso de Atletas			
Prueba 1500 metros			
Test de proyecciones			
Laboratorio Clínico			
Laboratorio de Desarrollo Físico	3	2	3
Laboratorio Cardiovascular			
Historia Clínica			
Prueba de Urea	1	3	1

2.4 Plan de Entrega

Una vez que el cliente culmina la elaboración de las historias de usuario, se comienza con la creación del Plan de Entregas. El mismo se hace con la intención de que los programadores obtengan una estimación de dichas historias en cuanto al nivel de detalle, o sea, para fijar el período de tiempo que se puede tardar en la implementación de cada una.

El Plan de Entrega posibilita la obtención de una clasificación teniendo en cuenta el riesgo que se corre a la hora de implementar la historia de usuario. Estos datos se almacenan en campos que permanecen vacíos en la historia de usuario, el responsable de llenar estos datos es únicamente el programador una vez que haya hecho el análisis requerido de los mismos.

Tabla 12. Plan de Entregas.

Historias de usuario	Iteración 1 (27 de marzo/2010)	Iteración 2 (7 de abril/2010)	Iteración 3 (2 de mayo/2010)
Plan gráfico médico	1.0	Terminado	Terminado

Peso de atletas Prueba 1500 metros			
Test de proyecciones Laboratorio clínico Laboratorio de desarrollo físico Laboratorio cardiovascular	0.2	1.0	Terminado
Historia clínica Prueba de urea	0.1	0.5	Terminado

2.5 Conclusiones

Conociendo como se lleva a cabo actualmente la gestión de la información de las pruebas médicas en la Escuela Cubana de Judo Femenino se hizo una propuesta del sistema. Se especificaron los usuarios que están relacionados con su utilización y funcionamiento. Se realizó una descripción de las historias de usuario precisando por el cliente la prioridad de cada una, definiendo así el orden en el que serán implementadas las mismas. Se cuenta con nueve historias de usuarios las cuales serán implementadas en tres iteraciones.

Capítulo 3. Implementación y pruebas.

3.1 Introducción

En este capítulo se definen las tareas generadas a partir del desarrollo de las historias de usuarios durante las tres iteraciones planificadas, se puntualizan los artefactos (tarjetas CRC) que dirigen la descripción del sistema y las pruebas de aceptación realizadas.

3.2 Diseño del sistema

La metodología XP no requiere la descripción del sistema por medio de diagramas de clase utilizando notación UML, sino que se guía por técnicas como las tarjetas CRC (Contenido, Responsabilidad y Colaboración). Esto no implica que no se utilicen los diagramas para obtener una mejor visión y comunicación entre el equipo de trabajo, siempre y cuando su complejidad no sea alta y defina información importante.

3.2.1 Tarjetas CRC

Las características más sobresalientes de las tarjetas CRC son su simpleza y ductilidad. Una tarjeta CRC no es más que una ficha de papel o cartón que representa a una entidad del sistema (Casas, 2007).

Estas tarjetas se utilizan para estructurar las clases y a su vez definir las responsabilidades sobre las mismas, así como la simulación de escenarios en el sistema.

Tabla 13. Modelo de Tarjeta CRC.

Clase	
Responsabilidades	Colaboradores

3.2.1.1 Tarjetas CRC del sistema

El sistema a desarrollar dispone de diversos objetos, por lo que se definen las siguientes clases:

- Médico

- Plan gráfico médico
- Peso diario
- Prueba de 1500 metros
- Test de proyecciones
- Laboratorio clínico
- Laboratorio de desarrollo físico
- Laboratorio cardiovascular
- Prueba de urea
- Historia clínica
- Reportes

A continuación se definen las tarjetas CRC del sistema:

Tabla 14. Tarjeta CRC clase Usuarios.

Médico	
Usuarios que interactúan con el sistema	

Tabla 15. Tarjeta CRC clase Plan gráfico médico.

Plan gráfico médico	
Consultar plan gráfico médico	Plan gráfico

Tabla 16. Tarjeta CRC clase Peso diario.

Peso diario	
Insertar peso diario	Médico
Modificar peso diario	

Consultar peso diario	
Consultar reporte de peso diario	
Graficar peso diario de atleta	

Tabla 17. Tarjeta CRC clase Prueba de 1500 metros.

Prueba de 1500 metros	
Insertar prueba de 1500 metros	Médico
Modificar prueba de 1500 metros	Plan gráfico médico
Consultar prueba de 1500 metros	
Consultar reporte de prueba de 1500 metros	

Tabla 18. Tarjeta CRC clase Test de proyecciones.

Test de proyecciones	
Insertar test de proyecciones	Médico
Modificar test de proyecciones	Plan gráfico médico
Consultar test de proyecciones	
Consultar reporte de test de proyecciones	

Tabla 19. Tarjeta CRC clase Laboratorio clínico.

Laboratorio clínico	
Insertar laboratorio clínico	Médico
Modificar laboratorio clínico	Plan gráfico médico
Consultar laboratorio clínico	

Consultar reporte de laboratorio clínico	
--	--

Tabla 20. Tarjeta CRC clase Laboratorio de desarrollo físico.

Laboratorio de desarrollo físico	
Insertar laboratorio de desarrollo físico	Médico
Modificar laboratorio de desarrollo físico	Plan gráfico médico
Consultar laboratorio de desarrollo físico	
Consultar reporte de laboratorio de desarrollo físico	

Tabla 21. Tarjeta CRC clase Laboratorio cardiovascular.

Laboratorio cardiovascular	
Insertar laboratorio cardiovascular	Médico
Modificar laboratorio cardiovascular	Plan gráfico médico
Consultar laboratorio cardiovascular	
Consultar reporte de laboratorio cardiovascular	

Tabla 22. Tarjeta CRC clase Prueba de urea.

Prueba de urea	
Insertar prueba de urea	
Modificar prueba de urea	Médico
Consultar prueba de urea	
Consultar reporte de prueba de urea	

Tabla 23. Tarjeta CRC clase Historia clínica.

Historia clínica	
Insertar historia clínica	Médico
Modificar historia clínica	
Consultar historia clínica	

3.2.2 Diagrama de Despliegue

La metodología XP plantea que para un mejor entendimiento de las tareas, flujos y métodos de desarrollo de las funcionalidades se pueden crear diagramas, siempre que su creación no implique mayor esfuerzo que la implementación del mismo. Siguiendo este principio se elaboró el diagrama de despliegue que permite apreciar de forma visual cómo se encuentran relacionados físicamente los componentes en la aplicación.

Es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre nodos de cómputo (Jacobson, 2000). El sistema implementado se encuentra ubicado en un servidor Web Apache y el mismo se conecta a una base de datos *PostgreSQL*.

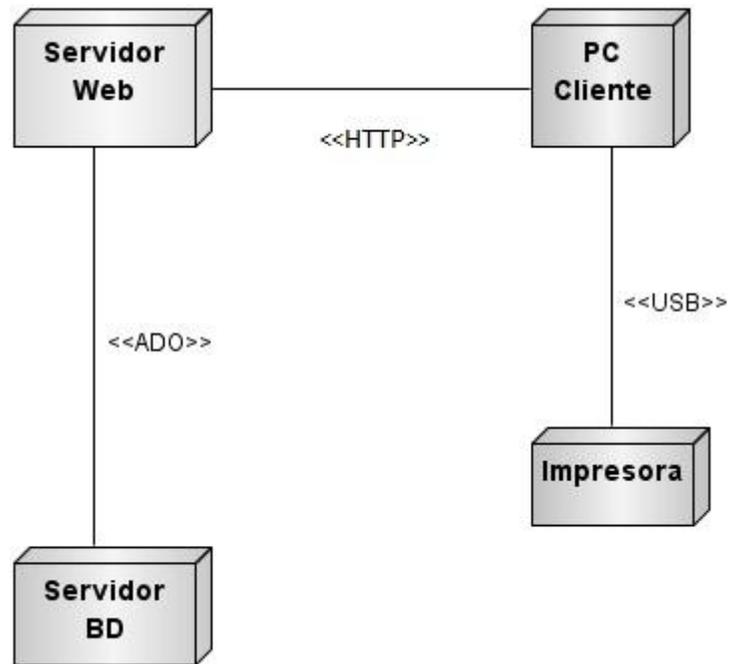


Figura. 3. Diagrama de Despliegue

3.2.3 Arquitectura del sistema

Debido a que el sistema que se desarrolló es basado en tecnología Web se utilizó el Modelo – Vista – Controlador.

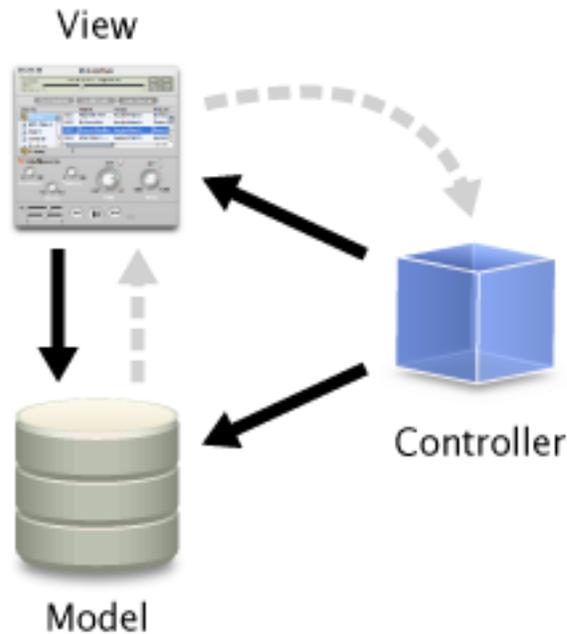


Figura 4. Modelo – Vista – Controlador (MVC)

El patrón conocido como Modelo – Vista – Controlador (MVC) separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes:

- Modelo: El modelo administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).
- Vista: Maneja la visualización de la información.
- Controlador: Interpreta las acciones del ratón y el teclado, informando al modelo y/o a la vista para que cambien según resulte apropiado.

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo independientemente de la representación visual. En aplicaciones Web, por otra parte, la separación entre la vista (el *browser*) y el controlador (los componentes del lado del servidor que manejan los requerimientos de HTTP) está mucho más taxativamente definida. (Reynoso, 2004).

3.3 Implementación

Las historias de usuario agrupadas en cada iteración se van implementando durante el transcurso de la iteración a la cual pertenecen. Al principio de cada iteración se lleva a cabo una revisión del plan de iteraciones y se modifica en caso de ser necesario. Como parte de este plan se descomponen las historias de usuarios en tareas de desarrollo, asignando a un grupo de desarrollo o una persona como responsable de su implementación. Estas tareas son para el uso estricto de los programadores por lo que pueden ser escritas en lenguaje técnico y no necesariamente entendible por el cliente.

Partiendo de la planificación realizada anteriormente se llevaron a cabo tres iteraciones de desarrollo sobre el sistema, permitiendo que al final de la última iteración se obtuviera un producto con todas las restricciones y características deseadas por el cliente. A continuación se detallan las tareas de desarrollo realizadas en cada una de las iteraciones.

3.3.1 Iteración 1

En esta iteración fueron planificadas las historias de usuarios de vital importancia para el cliente y las que definen la arquitectura base del sistema

Esta iteración tiene como objetivo darle cumplimiento a las historias de usuario que recogen las principales funcionalidades del sistema y representan un mayor valor para el cliente, pues con las mismas se conformará la base de la estructura del negocio.

3.3.1.1 Historias de usuarios planificadas para la primera iteración

Tabla 24. Historias de usuarios planificadas para la primera iteración.

Historia de usuario	Tiempo estimado (Semanas)	Tiempo real (Semanas)
Plan gráfico médico	0.6	0.4
Peso de atletas	0.8	0.8
Prueba de 1500 metros	0.8	0.8

3.3.1.2 Tareas de las historias de usuarios desarrolladas en la primera iteración

Luego de relacionar las historias de usuario que se les da cumplimiento en esta iteración se procede a la especificación de las principales tareas de desarrollo que se realizaron para cumplir el propósito de esta iteración. (Ver Anexo 1).

3.3.2 Iteración 2

Esta iteración está centrada en desarrollar una parte de las funcionalidades con prioridad media en el negocio. Está conformada por historias de usuario que no poseen un nivel de riesgo muy elevado para el programador.

3.3.2.1 Historias de usuarios planificadas para la segunda iteración

Tabla 25. Historias de usuarios planificadas para la segunda iteración.

Historia de usuario	Tiempo estimado (Semanas)	Tiempo real (Semanas)
Test de proyecciones	0.8	0.8
Laboratorio clínico	0.8	0.8
Laboratorio de desarrollo físico	0.8	0.8
Laboratorio cardiovascular	0.6	0.6

3.3.2.1 Tareas de las historias de usuarios desarrolladas en la segunda iteración

Luego de identificar las historias de usuario que se les da cumplimiento en esta iteración se procede a la especificación de las principales tareas de desarrollo que se realizaron para cumplir con las funcionalidades que se plantean en cada una de ellas. (Ver Anexo 2).

3.3.3 Iteración 3

En esta iteración se da continuidad a los objetivos de la iteración anterior, pues se enmarca en las restantes funcionalidades definidas con prioridad media para el cliente.

3.3.3.1 Historias de usuarios planificadas para la tercera iteración

Tabla 26. Historias de usuarios planificadas para la tercera iteración

Historia de usuario	Tiempo estimado (Semanas)	Tiempo real (Semanas)
Prueba de urea	0.6	0.6
Historia clínica	0.4	0.4

3.3.3.2 Tareas de las historias de usuarios desarrolladas en la iteración

Luego de plantear las historias de usuario que se les da cumplimiento en esta iteración se procede a la especificación de las principales tareas de desarrollo que se realizaron para cumplir con las funcionalidades que se plantean en cada una de ellas. (Ver Anexo 3).

3.3.4 Tareas de desarrollo genéricas

Aquí se exponen las tareas que son desarrolladas para cada una de las historias de usuarios debido a que están presentes en cada funcionalidad descrita por las mismas.

3.3.4.1 Tareas genéricas de las historias de usuarios

Tabla 27. Tareas genéricas de las historias de usuarios

Historia de usuario	Tareas genéricas
1, 2, 3, 4, 5, 6, 7, 8, 9	Sincronizar con la base de datos
	Exportar a PDF

3.3.4.2 Descripción de las tareas genéricas

Tabla 28. Sincronizar los datos con la base de datos.

Tarea	
Número de la tarea: 1	Número de HU: 1, 2, 3, 4, 5, 6, 7, 8, 9
Nombre de la tarea: Sincronizar los datos con la base de datos.	
Tipo de tareas: Diseño y desarrollo (Desarrollo/ Corrección/ Mejora / Investigación)	Puntos estimados: 0.4
Programador(es) responsable(s): Francisco Rodríguez González – Omar Mora Pérez	
Descripción Esta tarea define la interacción de consulta y petición de datos a la base de datos del sistema.	

Tabla 29. Exportar a PDF.

Tarea	
Número de la tarea: 2	Número de HU: 1, 2, 3, 4, 5, 6, 7, 8, 9
Nombre de la tarea: Exportar a PDF.	
Tipo de tareas: Desarrollo	Puntos estimados: 0.2
Programador(es) responsable(s): Omar Mora Pérez – Francisco Rodríguez González	
Descripción Esta tarea está referida a la edición de los datos obtenidos del sistema en formato de documento PDF.	

3.4 Estándares de codificación

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y

mantenimiento del código (Fernández, 2008). La metodología XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo facilitando los cambios (Penades, 2008). Para la implementación del sistema desarrollado se siguieron normas y estándares desarrollados por el equipo que se relacionan a continuación.

3.4.1 Definiciones generales

Las definiciones se realizan de manera descriptiva, evitando las abreviaturas y los nombres cortos.

Clases:

Las clases comienzan con mayúscula al inicio de la palabra y en caso de estar conformada por palabras compuestas, la definición debe ser continua y cada palabra debe iniciar con mayúscula siguiendo el estilo determinado. Además especifican en cada caso a qué tipo de clase se refiere.

Ejemplos:

```
class ControladoraPrueba1500{
```

```
...
```

```
class ControladoraADOPrueba1500 {
```

Métodos:

Los métodos deben empezar con mayúsculas. En caso de estar conformados por palabras compuestas, cada letra inicial de las palabras se debe escribir con mayúscula y sin separación entre palabras.

Ejemplos:

```
public function GetListaPG()
```

```
...
```

```
public function SetListaPG($plista_pg)
```

Variables:

Las variables comenzarán con minúsculas, aquellas que sean compuestas se escriben de manera seguida y con las primeras letras de cada palabra en mayúsculas sin incluir la primera. En caso de que las palabras sean muy extensas se utilizan abreviaturas. Se debe declarar cada variable en una línea distinta.

Ejemplos:

```
private $listaAtletas;
```

```
...
```

```
private $conectar;
```

```
...
```

```
private $conectar;
```

```
private $host;
```

```
private $port;
```

3.4.2 Comentarios

Se utilizan comentarios para especificar funciones y esclarecer alguna operación específica de algún método. La utilización de los mismos no está presente en todo el código, solo en aquellos casos que sea necesario.

3.4.3 Sentencias

3.4.3.1 Sentencias simples.

Cada línea debe contener una sola sentencia.

Ejemplo:

```
$datos = new ControladoraADOPruebaLabCineantropometria();
```

```
$micros = $datos->ListaMicro($pg,$num,$meso);
```

```
$lista = array();
```

3.4.3.2 Sentencias compuestas

Deben estar indentadas a un nivel superior que el precedente. Todas las sentencias del tipo *if* o *for*, *while*, *do... while* deben tener llaves. En el caso de las sentencias que solo posean una sola sentencia en su interior se pueden omitir las llaves.

Ejemplos:

```
if($this->ValidarFechaAux($this->ConvertirFecha($fecha),$f))
    return 1;
return $this->ConvertirFecha($fecha);
...
if (count($lista) !=0)
    {
        for ($i = 0;$i < count($lista);$i++) {
            if($lista[$i]->GetCI() == $ci)
                return $lista[$i];
        }
    }
return '';
```

3.5 Pruebas

En las metodologías tradicionales, la fase de pruebas, incluyendo la definición de las mismas, es usualmente realizada sobre el final del proyecto, o sobre el final del desarrollo de cada módulo. La metodología XP propone un modelo inverso, en el que, lo primero que se escribe son las pruebas que el sistema debe pasar. Luego, el desarrollo debe ser el mínimo necesario para pasar las pruebas previamente definidas. (Joskowicz, 2008)

La metodología XP divide las pruebas en dos grupos: las pruebas unitarias que son desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación que

están destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida y esperada por el cliente.

3.5.1 Pruebas de aceptación

Las pruebas de aceptación son creadas en base a las historias de usuarios en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada.

Las pruebas de aceptación son consideradas como pruebas de caja negra. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. (Joskowicz, 2008)

Las pruebas de aceptación son creadas a partir de las HU. Durante una iteración la HU seleccionada en la planificación de iteraciones se convertirá en una prueba de aceptación. El cliente o usuario especifica los aspectos a testear cuando una HU ha sido correctamente implementada. Una HU puede tener más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento. (Escribano, 2002) Una historia de usuario no se puede considerar terminada hasta que no pase correctamente todas las pruebas de aceptación.

3.5.1.1 Pruebas de aceptación de la HU Plan gráfico médico

Tabla 30. Caso de Pruebas No. 1 de la HU Plan gráfico médico.

Caso de prueba de Aceptación	
Código: HU1_P1	Historia de Usuario: 1
Nombre: Plan gráfico médico.	
Descripción: Prueba para la funcionalidad de consultar la planificación de las pruebas médicas en el sistema.	
Condiciones de ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	

<p>Entradas/Pasos de ejecución:</p> <p>Se selecciona el nombre del plan gráfico y el macrosistema correspondiente al mismo que se desea consultar para ver en qué mesosistemas están planificadas las pruebas médicas.</p>
<p>Resultado esperado: Se muestra una tabla con los mesosistemas y las fechas en que están planificadas pruebas médicas.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 31. Caso de Pruebas No. 2 de la HU Plan gráfico médico.

Caso de prueba de Aceptación	
Código: HU1_P2	Historia de Usuario: 1
Nombre: Plan gráfico médico.	
Descripción: Prueba para la funcionalidad de exportar la planificación de las pruebas médicas consultadas a documento PDF.	
<p>Condiciones de ejecución:</p> <p>El usuario debe estar autenticado.</p> <p>Se debe haber consultado y mostrado un plan gráfico médico.</p>	
<p>Entradas/Pasos de ejecución:</p> <p>Se da clic en la foto referente al documento PDF.</p>	
Resultado esperado: Se muestra en la ventana la planificación de las pruebas médicas del macrosistema antes consultado.	
Evaluación de la prueba: Prueba satisfactoria.	

3.5.1.2 Pruebas de aceptación de la HU Peso de atletas

Tabla 32. Caso de Pruebas No. 1 de la HU Peso de atletas.

Caso de prueba de Aceptación	
Código: HU2_P1	Historia de Usuario: 2
Nombre: Peso de atletas	
Descripción: Prueba para la funcionalidad de realizar el pesaje diario de las atletas.	
Condiciones de ejecución: El usuario debe estar autenticado. Se utilizará un usuario con datos válidos.	
Entradas/Pasos de ejecución: Se selecciona la fecha del día en que se realiza el pesaje y se introducen los datos del pesaje a cada atleta.	
Resultado esperado: Se muestra un aviso de que se han introducido los datos correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 33. Caso de Pruebas No. 2 de la HU Peso de atletas.

Caso de prueba de Aceptación	
Código: HU2_P2	Historia de Usuario: 2
Nombre: Peso de atletas	
Descripción: Prueba para la funcionalidad de modificar el pesaje diario de las atletas.	
Condiciones de ejecución: El usuario debe estar autenticado.	

Debe de existir un pesaje ya realizado.
Entradas/Pasos de ejecución: Se selecciona la fecha del día en que desea cambiar los datos y se modifican los datos del pesaje ya existente a cada atleta.
Resultado esperado: Se muestra un aviso de que los datos fueron modificados correctamente.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 34. Caso de Pruebas No. 3 de la HU Peso de atletas.

Caso de prueba de Aceptación	
Código: HU2_P3	Historia de Usuario: 2
Nombre: Peso de aletas	
Descripción: Prueba para la funcionalidad de consultar el pesaje diario de las atletas.	
Condiciones de ejecución: El usuario debe estar autenticado. Debe de existir un pesaje ya realizado.	
Entradas/Pasos de ejecución: Se selecciona la fecha del día en que desea consultar los datos del pesaje ya existente de cada atleta.	
Resultado esperado: Se muestran los datos del pesaje realizado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 35. Caso de Pruebas No. 4 de la HU Peso de atletas.

Caso de prueba de Aceptación

Código: HU2_P4	Historia de Usuario: 2
Nombre: Peso de aletas	
Descripción: Prueba para la funcionalidad de graficar el pesaje diario de las atletas.	
Condiciones de ejecución: El usuario debe estar autenticado. Debe de existir un pesaje ya realizado.	
Entradas/Pasos de ejecución: Se selecciona el plan gráfico médico, el macrosistema, el mesosistema y el microsistema en los que se desean observar la variación en el pesaje y luego a la atleta a la que se desea consultar.	
Resultado esperado: Se muestra una gráfica con la variación de peso en ese período.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 36. Caso de Pruebas No. 5 de la HU Peso de atletas.

Caso de prueba de Aceptación	
Código: HU3_P5	Historia de Usuario: 3
Nombre: Peso de aletas	
Descripción: Prueba para la funcionalidad de obtención del reporte de los resultados del pesaje de las atletas en fechas anteriores.	
Condiciones de ejecución: El usuario debe estar autenticado. Debe existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se haya realizado el pesaje diario de las atletas.	
Entradas/Pasos de ejecución:	

Se selecciona la atleta a la que se le desea mostrar el reporte de los resultados del pesaje diario anteriormente.
Resultado esperado: Se muestran los resultados del pesaje de las atletas registradas en el sistema.
Evaluación de la prueba: Prueba satisfactoria.

3.5.1.3 Pruebas de aceptación de la HU Prueba de 1500 metros

Tabla 37. Caso de Pruebas No. 1 de la HU Prueba de 1500 metros.

Caso de prueba de Aceptación	
Código: HU3_P1	Historia de Usuario: 3
Nombre: Prueba de 1500 metros	
Descripción: Prueba para la funcionalidad de añadir resultados de la prueba de 1500 metros de las atletas.	
Condiciones de ejecución: El usuario debe estar autenticado. Debe existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan planificado pruebas médicas.	
Entradas/Pasos de ejecución: Se selecciona el plan gráfico médico, el macrosistema, el mesosistema y el microsistema en los que se planificaron pruebas médicas y la fecha en que se realizó la prueba de 1500 metros, luego se introducen los resultados al sistema.	
Resultado esperado: Se muestra un aviso de que se han introducido los datos correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 38. Caso de Pruebas No. 2 de la HU Prueba de 1500 metros.

Caso de prueba de Aceptación	
Código: HU3_P2	Historia de Usuario: 3
Nombre: Prueba de 1500 metros	
Descripción: Prueba para la funcionalidad de modificar los resultados de la prueba de 1500 metros de las atletas.	
Condiciones de ejecución: El usuario debe estar autenticado. Debe existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan realizado pruebas de 1500 metros.	
Entradas/Pasos de ejecución: Se selecciona el plan gráfico médico, el macrosistema, el mesosistema y el microsistema en los que se añadieron los datos de la prueba de 1500 metros y luego se modifican los resultados en el sistema.	
Resultado esperado: Se muestra un aviso de que se han modificado los datos correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 39. Caso de Pruebas No. 3 de la HU Prueba de 1500 metros.

Caso de prueba de Aceptación	
Código: HU3_P3	Historia de Usuario: 3
Nombre: Prueba de 1500 metros	
Descripción: Prueba para la funcionalidad de consultar los resultados de la prueba de 1500 metros de las atletas.	
Condiciones de ejecución:	

<p>El usuario debe estar autenticado.</p> <p>Debe existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan realizado pruebas de 1500 metros.</p>
<p>Entradas/Pasos de ejecución:</p> <p>Se selecciona el plan gráfico médico, el macrosistema, el mesosistema y el microsistema en los que se añadieron los datos de la prueba de 1500 metros y luego se modifican los resultados en el sistema.</p>
<p>Resultado esperado: Se muestran los datos de la prueba de 1500 metros registrada.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 40. Caso de Pruebas No. 4 de la HU Prueba de 1500 metros.

Caso de prueba de Aceptación	
Código: HU3_P4	Historia de Usuario: 3
Nombre: Prueba de 1500 metros	
Descripción: Prueba para la funcionalidad de obtención del reporte de los resultados de la prueba de 1500 metros de las atletas en períodos anteriores.	
<p>Condiciones de ejecución:</p> <p>El usuario debe estar autenticado.</p> <p>Debe existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan realizado pruebas de 1500 metros.</p>	
<p>Entradas/Pasos de ejecución:</p> <p>Se selecciona la atleta a la que se le desea mostrar el reporte de los resultados obtenidos en esta prueba anteriormente.</p>	
Resultado esperado: Se muestran los resultados de todas las pruebas de 1500 metros registradas	

en el sistema.

Evaluación de la prueba: Prueba satisfactoria.

3.5.1.4 Pruebas de aceptación de la HU Test de proyecciones

Tabla 41. Caso de Pruebas No. 1 de la HU Test de proyecciones.

Caso de prueba de Aceptación	
Código: HU4_P1	Historia de Usuario: 4
Nombre: Test de proyecciones	
Descripción: Prueba para la funcionalidad de añadir resultados de la prueba de proyecciones de las atletas.	
Condiciones de ejecución: El usuario debe estar autenticado. Debe existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan planificado pruebas médicas.	
Entradas/Pasos de ejecución: Se selecciona el plan gráfico médico, el macrosistema, el mesosistema y el microsistema en los que se planificaron pruebas médicas y la fecha en que se realizó el test de proyecciones, luego se introducen los resultados al sistema.	
Resultado esperado: Se muestra un aviso de que se han introducido los datos correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Para observar los casos de pruebas de las funcionalidades modificar, consultar y obtención de reporte del test de proyecciones ver Anexo 4.

3.5.1.5 Pruebas de aceptación de la HU Laboratorio clínico

Tabla 42. Caso de Pruebas No. 1 de la HU Laboratorio clínico.

Caso de prueba de Aceptación	
Código: HU5_P1	Historia de Usuario: 5
Nombre: Laboratorio clínico	
Descripción: Prueba para la funcionalidad de añadir resultados de las pruebas del laboratorio clínico de las atletas.	
Condiciones de ejecución: El usuario debe estar autenticado. Debe existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan planificado pruebas médicas.	
Entradas/Pasos de ejecución: Se selecciona el plan gráfico médico, el macrosistema, el mesosistema y el microsistema en los que se planificaron pruebas médicas, luego se introducen los resultados de las pruebas de laboratorio clínico al sistema.	
Resultado esperado: Se muestra un aviso de que se han introducido los datos correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Para observar los casos de pruebas de las funcionalidades modificar, consultar y obtención de reporte de los resultados de las pruebas de laboratorio clínico ver Anexo 5.

3.5.1.6 Pruebas de aceptación de la HU Laboratorio de desarrollo físico o cineantropométrico.

Tabla 43. Caso de Pruebas No.1 de la HU Laboratorio de desarrollo físico.

Caso de prueba de Aceptación	
Código: HU6_P1	Historia de Usuario: 6
Nombre: Laboratorio de desarrollo físico	
Descripción: Prueba para la funcionalidad de añadir los resultados de las pruebas del laboratorio de desarrollo físico de las atletas en el sistema.	
Condiciones de ejecución: El usuario debe estar autenticado. Debe existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan planificado pruebas médicas.	
Entradas/Pasos de ejecución: Se selecciona el plan gráfico médico, el macrosistema, el mesosistema y el microsistema en los que se planificaron pruebas médicas, luego se introducen los resultados de las pruebas de laboratorio de desarrollo físico realizadas.	
Resultado esperado: Se muestra un aviso de que se han introducido los datos correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Para observar los casos de pruebas de las funcionalidades modificar, consultar y obtención de reporte de los resultados de las pruebas de laboratorio de desarrollo físico ver Anexo 6.

3.5.1.7 Pruebas de aceptación de la HU Prueba de laboratorio cardiovascular

Tabla 44. Caso de Pruebas No.1 de la HU Laboratorio cardiovascular.

Caso de prueba de Aceptación	
Código: HU7_P1	Historia de Usuario: 7
Nombre: Laboratorio cardiovascular	
Descripción: Prueba para la funcionalidad de añadir los resultados de las pruebas del laboratorio cardiovascular de las atletas en el sistema.	
Condiciones de ejecución: El usuario debe estar autenticado. Debe existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan planificado pruebas médicas.	
Entradas/Pasos de ejecución: Se selecciona el plan gráfico médico, el macrosistema, el mesosistema y el microsistema en los que se planificaron pruebas médicas, luego se introducen los resultados de las pruebas de laboratorio cardiovascular realizadas.	
Resultado esperado: Se muestra un aviso de que se han introducido los datos correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Para observar los casos de pruebas de las funcionalidades modificar, consultar y obtención de reporte de los resultados de las pruebas de laboratorio cardiovascular ver Anexo 6.

3.5.1.8 Pruebas de aceptación de la HU Historia clínica

Tabla 45. Caso de Pruebas No.1 de la HU Historia clínica.

Caso de prueba de Aceptación

Código: HU8_P1	Historia de Usuario: 8
Nombre: Historia clínica	
Descripción: Prueba para la funcionalidad de añadir los datos necesarios para la confección de la historia clínica individual de cada atleta en el sistema.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Entradas/Pasos de ejecución: Se selecciona la atleta a la que se le desea confeccionar la historia clínica y se identifican los principales padecimientos, lesiones y enfermedades.	
Resultado esperado: Se muestra un aviso de que se han introducido los datos correctamente al sistema.	
Evaluación de la prueba: Prueba satisfactoria.	

Para observar los casos de pruebas de las funcionalidades modificar y consultar los datos de las historias de usuario de las atletas ver Anexo 7.

3.5.1.9 Pruebas de aceptación de la HU Prueba de Urea

Tabla 46. Caso de Pruebas No.1 de la HU Prueba de urea.

Caso de prueba de Aceptación	
Código: HU9_P1	Historia de Usuario: 9
Nombre: Prueba de urea	
Descripción: Prueba para la funcionalidad de añadir el resultado de la prueba de urea de las atletas.	
Condiciones de ejecución:	

El usuario debe estar autenticado.

Se utilizará un usuario con datos válidos.

Entradas/Pasos de ejecución:

Se selecciona la fecha del día en que se realiza la prueba de urea y se introducen los resultados de cada atleta.

Resultado esperado: Se muestra un aviso de que se han introducido los resultados correctamente.

Evaluación de la prueba: Prueba satisfactoria.

Para observar los casos de pruebas de las funcionalidades modificar, consultar y obtención de reporte de los resultados de las pruebas de urea aplicadas a las atletas ver Anexo 8.

3.6 Conclusiones

En este capítulo se elaboró el modelo necesario para llevar a cabo la implementación del sistema mediante la descripción de los estándares de codificación y la arquitectura utilizada. Se plantearon y describieron las tareas de desarrollo para dar cumplimiento a las funcionalidades abordadas por las historias de usuario. Se definió por medio del diagrama de despliegue la distribución física mediante la cual funcionará la aplicación y se detallaron las pruebas de aceptación que determinan la confianza y seguridad de las funcionalidades del sistema para el cliente.

Conclusiones generales

Con el desarrollo de esta investigación se realizó un estudio de los principales sistemas que gestionan la información médica de los deportistas además del nivel de aceptación por parte de los usuarios. Se demostró la necesidad de construir un sistema capaz de informatizar todo el proceso de gestión de la información de los resultados de las pruebas médicas aplicadas a las atletas en la Escuela Cubana de Judo Femenino y de esta forma lograr una mayor seguridad, facilidad y rapidez en el manejo de la información.

El desarrollo del trabajo utilizando como guía la metodología XP permitió una mejor organización dentro del equipo de trabajo garantizando un rendimiento máximo por parte de cada integrante además de la creación del software en un tiempo fiable que responde en todo momento a los intereses del cliente. Se confeccionó una documentación a través de todo el ciclo de desarrollo del producto para facilitar la comprensión del sistema por parte de otros desarrolladores en futuras versiones.

Con este trabajo se presentó una aplicación Web que gestiona los resultados de las pruebas médicas realizadas a las judocas y genera una historia clínica con los principales padecimientos, lesiones y enfermedades de las mismas.

Recomendaciones

Una vez concluido el desarrollo de este trabajo se recomienda:

- ✓ Agregar nuevas funcionalidades al sistema de acuerdo con las necesidades futuras del cliente tanto para la gestión de los datos como para garantizar la integridad y seguridad de los mismos.
- ✓ La utilización de esta experiencia como material de estudio para el desarrollo de aplicaciones similares por parte de otros desarrolladores.
- ✓ La utilización de la metodología *Extreme Programming* en los procesos de desarrollo de software pequeños permitiendo la creación de un producto en un tiempo fiable y que responde en todo momento a los intereses del cliente.
- ✓ Realizar el Manual de Usuario del sistema.

Referencias Bibliográficas

1. PALOMEQUE, Horacio. Bodimetrix. Antropometría al Máximo Nivel [en línea] (2005). Disponible en: <http://www.bodymetrix.com.ar>
2. PALOMEQUE, Horacio. X-Training Fussion®. Software para planificación, periodización y control del entrenamiento deportivo. [en línea] (2005). Disponible en: <http://www.x-trainingfussion.com.ar>
3. PALOMEQUE, Horacio. X-Medalist®. Nuevo software para entrenadores personales. [en línea] (2005). Disponible en: <http://www.x-medalist.com.ar>
4. MOLPECERES, Alberto. Procesos de desarrollo: RUP, XP y FDD. (2002) Disponible en: <http://www.willydev.net/descargas/Articulos/General/cualxpfddrup.PDF>
5. BECK, K. (1999). Extreme Programming Explained. Embrace Change (Una explicación de la programación extrema. Aceptar el cambio). s.l. : Pearson Education. Disponible en: <http://www.mip.sdu.dk/~brianj/Extreme%20Programming%20Explained%20-%20Kent%20Beck;%20Addison-Wesley,%201999.pdf>
6. SCHENONE, Marcelo y Villagra, Sergio. (2004). Diseño de una Metodología Ágil de Desarrollo de Software. Facultad de Ingeniería. Buenos Aires (Universidad). Disponible en: <http://materias.fi.uba.ar/7500/schenone-tesisdegradoingenieriainformatica.pdf>
7. LETELIER, Patricio y PENADÉS, M^a Carmen. "Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)". (2008). [en línea]. Disponible en: <http://www.willydev.net/InsiteCreation/v1.0/descargas/masyxp.pdf>
8. MOLPECERES, Alberto. Procesos de desarrollo: RUP, XP y FDD. (2002) Disponible en: <http://www.willydev.net/descargas/Articulos/General/cualxpfddrup.PDF>
9. LAFUENTE, Guillermo Javier. GIDISWEB. UML Unified Modeling Lenguaje. [En línea] (1998). <http://gidis.ing.unlpam.edu.ar/personas/glafuente/uml/uml.html>.
10. MANZANEDO, Miguel Ángel, F. J. G. P. (1999). "Guía Rápida de Aprendizaje del Lenguaje Java." Disponible en: <http://pisuerga.inf.ubu.es/lsi/Invest/Java/Tuto.Oct98/index.htm>.

11. HINOSTROZA, R. R. (2005). "LinuxCentro.net. Características de PHP." Disponible en: <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.
12. ARAMBILLETE, Wilman. WebTaller. Características de JSP. [En línea] 2003. <http://www.Webtaller.com>.
13. ALVAREZ, Gonzalo. Departamento de Tratamiento de la Información y Codificación. *Que es Java*. [En línea] 2003. Disponible en: <http://www.iec.csic.es>.
14. Valdelli, I. (2006). "HTMLPOINT.com. CURSO JAVASCRIPT." from <http://www.htmlpoint.com/javascript/corso/index.html>
15. ALVAREZ, Miguel Angel. desarrolloweb.com. Manual de CSS, hojas de estilo. [En línea] (2000) <http://www.desarrolloweb.com/manuales/manual-css-hojas-de-estilo.html>.
16. GUTIÉRREZ, Juan. Tutorial de Eclipse. (2004) Disponible en: http://www.uv.es/~jgutierrez/MySQL_Java/TutorialEclipse.pdf
17. HERNANDIS, J. A. Why Visual Paradigm for UML?. (2005). Disponible en: <http://www.visual-paradigm.com/product/vpuml/>
18. TABOADA, Alberto. UML Unified Modeling Language [en línea]. Edición 1.0. [citado 15 noviembre 2010]. Disponible en: <http://maestroquake.iespana.es/archivos/FundamentosUML.pdf>
19. ALIAGA, Antonio; MIANI, Marcos. Trabajo investigativo sobre postgresQL. (2008). [En línea]. Disponible en: www.iessanvicente.com/colaboraciones/postgreSQL.pdf
20. MANUEL, José ¿Qué es MySQL? [En línea]. Revista: Especialistas Web. Editado: 16-08-2005 [Consultado: 5 de noviembre del 2010]. Disponible en: <http://www.espepestudio.com/articulo/desarrollo-web/bases-de-datos-mysql/Que-es-MySQL.htm>
21. JOSKOWICZ, José. (2008). "Reglas y Prácticas en eXtreme Programming". (2008). Disponible en: <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
22. ESCRIBANO, G. F. Introducción a Extreme Programming, 2002. Disponible en: <http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-XP.pdf>

23. FERNÁNDEZ, Carlos. Estándar para codificación en lenguaje c ++. (2008). Publicación Web “Coordinación docente de lenguajes de programación”. Disponible en:
<http://progra.iteso.mx/estandares/estandar%20codificacion%20c%2B%2B/estandarcodificacion.pdf>
24. JACOBSON, G. BOOCH, J, RUMBAUGH. El Proceso Unificado de Desarrollo de Software. (2000). Disponible en: <http://bibliodoc.uci.cu/pdf/8478290362.pdf>
25. CASAS, Sandra; REINAGA, Héctor. Identificación y Modelado de Aspectos Tempranos dirigido por Tarjetas de Responsabilidades y Colaboraciones. (2007). [en línea]
Disponible en: <http://espanol.oocities.com/profeprog2/INVPAPER25.pdf>
26. REYNOSO, Carlos. KICCILLOF, Nicolas. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft. Versión 1.0. (2004). UNIVERSIDAD DE BUENOS AIRES. Disponible en:
<http://www.willydev.net/descargas/prev/Estiloypatron.pdf>

Bibliografía

- MOLPECERES, Alberto. Procesos de desarrollo: RUP, XP y FDD. Procesos de desarrollo: RUP, XP y FDD. (2002) Disponible en: <http://www.willydev.net/descargas/Articulos/General/cualxpfdrrup.PDF>.
- LAFUENTE, Guillermo Javier. GIDISWEB. UML Unified Modeling Language. [en línea] (1998). Disponible en: <http://gidis.ing.unlpam.edu.ar/personas/glafuente/uml/uml.html>.
- ARAMBILLETE, Wilman. WebTaller. Características de JSP. [en línea] (2003). Disponible en: <http://www.Webtaller.com>.
- ALVAREZ, Gonzalo. Departamento de Tratamiento de la Información y Codificación. Que es Java. [en línea] (1997). Disponible en: <http://www.iec.csic.es>.
- ALVAREZ, Miguel Angel. Revista Digital: Desarrolloweb.com. Manual de CSS, hojas de estilo. [en línea] (2000) Disponible en: <http://www.desarrolloweb.com/manuales/manual-css-hojas-de-estilo.html>.
- RATIONAL ROSE ENTERPRISE RATIONAL.COM [en línea] Disponible en: <http://www.rational.com.ar/herramientas/roseenterprise.html>
- OÑA, A. Martínez, M. & Moreno, F. Descripción de un sistema informatizado de procesamiento automático para la optimización del rendimiento deportivo basado en el control de la información. Revista Motricidad (1994) Vol 1. p.57-69. Disponible en: <http://www.cienciadeporte.com/motricidad/1/art4.pdf>
- GARCÍA, Fara O; Aguiar, Xenia M.; González, Luis A; Morales, Antonio; Sánchez, Alberto. Automatización de la planificación del entrenamiento deportivo en diferentes deportes. Publicación científica. Facultad de Cultura Física Villa Clara. Texto íntegro: Biblioteca Facultad de Cultura Física Villa Clara
- LEÓN, Sofía; GIRÓN, José. Metodología para la aplicación y realización de pruebas pedagógicas y médicas en el deporte de alta calificación. (2005). Disponible en: <http://delegaciongranadinatkd.iespana.es/articulos.htm>

- MANZANEDO, Miguel Ángel, F. J. G. P. (1999). "Guía Rápida de Aprendizaje del Lenguaje Java." Disponible en: <http://pisuerga.inf.ubu.es/lsi/Invest/Java/Tuto.Oct98/index.htm>.
- SCHENONE, Marcelo; Villagra, Sergio. (2004). Diseño de una Metodología Ágil de Desarrollo de Software. Facultad de Ingeniería. Buenos Aires (Universidad).
- JOSKOWICZ, José. (2008). "Reglas y Prácticas en eXtreme Programming". Disponible en: <http://ie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
- PENADÉS, M^a Carmen; Letelier, Patricio. (2008). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [en línea] 29 de Abril de 2008. [Citado el: 10 de Marzo de 2010.] Disponible en: <http://www.willydev.net/descargas/masyxp.pdf>.
- SOLÍS, M. C. (2003). "Una explicación de la programación extrema (XP)." from <http://www.willydev.net/InsiteCreation/v1.0/descargas/prev/explicaxp.pdf>.
- KRUCHTEN, P. 2000. The Rational Unified Process: An Introduction. [En línea]. Disponible en: http://books.google.com/books?id=RYCMx6o47pMC&dq=The+Rational+Unified+Process:+An+Introduction+2002&printsec=frontcover&source=bn&hl=es&ei=4sfIS7rFFoagIAfU4YTqCQ&sa=X&oi=book_result&ct=result&resnum=4&ved=0CB8Q6AEwAw#v=onepage&q=The%20Rational%20Unified%20Process%3A%20An%20Introduction%202002&f=false
- TABOADA, Alberto. UML Unified Modeling Language [en línea]. Edición 1.0. [citado 15 noviembre 2010]. Disponible en: <http://maestroquake.iespana.es/archivos/FundamentosUML.pdf>
- CASTILLO, A. D. (2007). "Manual PHP. ¿Qué es PHP? una breve introducción." Disponible en: <http://www.lawebera.es/manuales/php/1.php>.
- ALIAGA, Antonio; Miani, Marcos. Trabajo investigativo sobre postgreSQL. (2008). [en línea]. Disponible en: www.iessanvicente.com/colaboraciones/postgreSQL.pdf

Glosario de términos

Test pedagógico: Es la prueba que se le realiza a las atletas para conocer su condición física y el nivel de preparación que tienen.

Plan gráfico: Proceso que contiene de forma organizada toda la planificación de las actividades que se le van a realizar a los atletas y el volumen y la intensidad con que se les van a aplicar para que alcancen la forma deportiva antes de presentarse a la competencia fundamental de ese periodo.

Plan gráfico médico: Proceso que contiene de forma organizada toda la planificación de las pruebas médicas que se le van a realizar a las atletas.

Macrosistemas: Engloban el total de objetivos marcados en un proceso completo de entrenamiento. Están formados por mesosistemas.

Mesosistemas: Representan etapas del proceso global de entrenamiento. Están orientados a lograr el desarrollo de una cualidad u objetivo parcial de todo el proceso. Su organización interna se realiza a base de microsistemas.

Microsistema: Representa una semana.

Plug-in: Son programas que se agregan a otro ya existente para ofrecer nuevas funcionalidades. Estos programas no funcionarían de forma independiente, necesitan que ya esté instalado otro programa concreto en el ordenador donde se instalan.

Urea: Es una sustancia con alto contenido en nitrógeno que se produce cuando el cuerpo metaboliza las proteínas. Se produce en el hígado y el riñón es el encargado de eliminarlo del cuerpo a través de la orina. Normalmente, se determina su concentración en una muestra de sangre. El objetivo de esta prueba diagnóstica es determinar si los riñones funcionan normalmente, si una enfermedad renal existente ha empeorado, para vigilar el tratamiento de una enfermedad renal o para determinar si una persona está deshidratada o no.

PDF: (acrónimo del inglés *portable document format*, formato de documento portátil) es un formato de almacenamiento de documentos, desarrollado por la empresa *Adobe Systems*.

Cineantropometría: Estudio de la composición corporal en su función y estructura. Mediciones corporales, talla, peso.

Servlets: Son módulos java que sirven para extender las capacidades de los servidores web.

Kernel: Es el software que constituye el núcleo del sistema operativo, dónde se realizan las funcionalidades básicas como la gestión de procesos, la gestión de memoria y de entrada / salida.

Anexo No 1.

Tareas de desarrollo de la primera iteración.

Tabla 47. Crear interfaz visual de presentación del plan gráfico médico.

Tarea	
Número de la tarea: 1	Número de HU: 1
Nombre de la tarea: Crear interfaz visual de presentación del plan gráfico médico	
Tipo de tareas: Diseño	Puntos estimados: 0.2
Fecha de inicio: 15 de Marzo del 2010	Fecha Fin: 15 de Marzo del 2010
Programador(es) responsable(s): Francisco Rodríguez González – Omar Mora Pérez	
<p>Descripción</p> <p>Esta tarea define el diseño de la pantalla que mostrará la información del plan gráfico médico de la planificación de las pruebas al usuario. Se crean las tablas y formularios para mostrar la información correspondiente. Se configuran los permisos para que el usuario pueda ver la información pero no modificarla.</p>	

Tabla 48. Mostrar plan gráfico medico.

Tarea	
Número de la tarea: 2	Número de HU: 1
Nombre de la tarea: Mostrar plan gráfico medico	
Tipo de tareas: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 16 de Marzo del 2010	Fecha Fin: 16 de Marzo del 2010
Programador(es) responsable(s): Francisco Rodríguez González – Omar Mora Pérez	
<p>Descripción</p>	

Una vez que se seleccione el nombre del plan gráfico y el macrosistema donde se planificaron pruebas médicas, se mostrarán en pantalla las fechas correspondientes a la planificación de las pruebas en cada mesosistema y microsistema.

Tabla 49. Crear interfaz visual de gestión del pesaje de las atletas.

Tarea	
Número de la tarea: 1	Número de HU: 2
Nombre de la tarea: Crear interfaz visual de gestión del pesaje de las atletas	
Tipo de tareas: Diseño	Puntos estimados: 0.4
Fecha de inicio: 17 de Marzo del 2010	Fecha Fin: 18 de Marzo del 2010
Programador(es) responsable(s): Francisco Rodríguez González – Omar Mora Pérez	
<p>Descripción</p> <p>Esta tarea define el diseño de las pantallas que mostrarán los datos de la gestión (insertar, modificar y consultar) de la información del pesaje de las judocas al usuario. Se crean las tablas y formularios para mostrar la información correspondiente y para introducir los datos correspondientes en los casos necesarios.</p>	

Tabla 50. Gestionar la información del pesaje de las judocas en el sistema.

Tarea	
Número de la tarea: 2	Número de HU: 2
Nombre de la tarea: Gestionar la información del pesaje de las judocas en el sistema.	
Tipo de tareas: Desarrollo (Desarrollo/ Corrección/ Mejora / Investigación)	Puntos estimados: 0.6
Fecha de inicio: 19 de Marzo del 2010	Fecha Fin: 23 de Marzo del 2010

Programador(es) responsable(s): Omar Mora Pérez – Francisco Rodríguez González
<p>Descripción</p> <p>Esta tarea está referida a la inserción, modificación y consulta de los datos del pesaje de las judocas en el sistema, así como la visualización gráfica de los mismos. Se definen los permisos de modificación y escritura en cada campo según se requiera. Se mostrará en pantalla la información correspondiente al pesaje diario.</p>

Tabla 51. Crear interfaz visual de gestión de la prueba de 1500 metros de las judocas

Tarea	
Número de la tarea: 1	Número de HU: 3
Nombre de la tarea: Crear interfaz visual de gestión de la prueba de 1500 metros de las judocas	
Tipo de tareas: Diseño	Puntos estimados: 0.4
Fecha de inicio: 26 de Marzo del 2010	Fecha Fin: 27 de Marzo del 2010
Programador(es) responsable(s): Francisco Rodríguez González – Omar Mora Pérez	
<p>Descripción</p> <p>Esta tarea define el diseño de las pantallas que mostrarán los datos de la gestión (insertar, modificar y consultar) de la información de la prueba de 1500 metros realizada por las judocas al usuario. Se crean las tablas y formularios para mostrar la información correspondiente y para introducir los datos correspondientes en los casos necesarios.</p>	

Tabla 52. Gestionar la información de la prueba de 1500 metros de las judocas

Tarea	
Número de la tarea: 2	Número de HU: 3
Nombre de la tarea: Gestionar la información de la prueba de 1500 metros de las judocas	

Tipo de tareas: Desarrollo	Puntos estimados: 0.6
Fecha de inicio: 28 de Marzo del 2010	Fecha Fin: 30 de Marzo del 2010
Programador(es) responsable(s): Omar Mora Pérez – Francisco Rodríguez González	
Descripción Esta tarea está referida a la inserción, modificación y consulta de los datos de la prueba de 1500 metros de las judocas en el sistema, así como la visualización gráfica de los mismos. Se definen los permisos de modificación y escritura de cada campo según se requiera. Se mostrará en pantalla la información correspondiente al pesaje diario.	

Anexo No 2.

Tareas de desarrollo de la segunda iteración.

Tabla 53. Crear interfaz visual de gestión del test de proyecciones de las judocas

Tarea	
Número de la tarea: 1	Número de HU: 4
Nombre de la tarea: Crear interfaz visual de gestión del test de proyecciones de las judocas	
Tipo de tareas: Diseño	Puntos estimados: 0.2
Fecha de inicio: 31 de Marzo del 2010	Fecha Fin: 31 de Marzo del 2010
Programador(es) responsable(s): Francisco Rodríguez González – Omar Mora Pérez	
<p>Descripción</p> <p>Esta tarea define el diseño de las pantallas que mostrarán los datos de la gestión (insertar, modificar y consultar) de la información del test de proyecciones realizado por las judocas al usuario. Se crean las tablas y formularios para mostrar la información correspondiente y para introducir los datos correspondientes en los casos necesarios.</p>	

Tabla 54. Gestionar la información del test de proyecciones de las judocas en el sistema

Tarea	
Número de la tarea: 2	Número de HU: 4
Nombre de la tarea: Gestionar la información del test de proyecciones de las judocas en el sistema.	
Tipo de tareas: Desarrollo (Desarrollo/ Corrección/ Mejora / Investigación)	Puntos estimados: 0.6
Fecha de inicio: 1 de Abril del 2010	Fecha Fin: 5 de Abril del 2010

Programador(es) responsable(s): Omar Mora Pérez – Francisco Rodríguez González
<p>Descripción</p> <p>Esta tarea está referida a la inserción, modificación y consulta de los datos del test de proyecciones de las judocas en el sistema, así como la visualización gráfica de los mismos. Se definen los permisos de modificación y escritura de cada campo según se requiera. Se mostrará en pantalla la información correspondiente al pesaje diario.</p>

Tabla 55. Crear interfaz visual de gestión de las pruebas de laboratorio clínico de las judocas.

Tarea	
Número de la tarea: 1	Número de HU: 5
Nombre de la tarea: Crear interfaz visual de gestión de las pruebas de laboratorio clínico de las judocas.	
Tipo de tareas: Diseño	Puntos estimados: 0.2
Fecha de inicio: 6 de Abril del 2010	Fecha Fin: 6 de Abril del 2010
Programador(es) responsable(s): Francisco Rodríguez González – Omar Mora Pérez	
<p>Descripción</p> <p>Esta tarea define el diseño de las pantallas que mostrarán los datos de la gestión (insertar, modificar y consultar) de la información de las pruebas de laboratorio clínico realizado a las judocas al usuario. Se crean las tablas y formularios para mostrar la información correspondiente y para introducir los datos correspondientes en los casos necesarios.</p>	

Tabla 56. Gestionar la información del test de proyecciones de las judocas en el sistema.

Tarea	
Número de la tarea: 2	Número de HU: 5
Nombre de la tarea: Gestionar la información del test de proyecciones de las judocas en el	

sistema.	
Tipo de tareas: Desarrollo	Puntos estimados: 0.6
Fecha de inicio: 7 de Abril del 2010	Fecha Fin: 9 de Abril del 2010
Programador(es) responsable(s): Omar Mora Pérez – Francisco Rodríguez González	
Descripción Esta tarea está referida a la inserción, modificación y consulta de los datos de las pruebas de laboratorio clínico de las judocas en el sistema, así como la visualización gráfica de los mismos. Se definen los permisos de modificación y escritura de cada campo según se requiera. Se mostrará en pantalla la información correspondiente al pesaje diario.	

Anexo No 3.

Tareas de desarrollo de la tercera iteración.

Tabla 57. Crear interfaz visual de gestión de la prueba de urea de las judocas

Tarea	
Número de la tarea: 1	Número de HU: 8
Nombre de la tarea: Crear interfaz visual de gestión de la prueba de urea de las judocas.	
Tipo de tareas: Diseño	Puntos estimados: 0.2
Fecha de inicio: 21 de Abril del 2010	Fecha Fin: 21 de Abril del 2010
Programador(es) responsable(s): Francisco Rodríguez González – Omar Mora Pérez	
<p>Descripción</p> <p>Esta tarea define el diseño de las pantallas que mostrarán los datos de la gestión (insertar, modificar y consultar) de la información de la prueba de urea realizada a las judocas al usuario. Se crean las tablas y formularios para mostrar la información correspondiente y para introducir los datos correspondientes en los casos necesarios.</p>	

Tabla 58. Gestionar la información de la prueba de urea de las judocas en el sistema

Tarea	
Número de la tarea: 2	Número de HU: 8
Nombre de la tarea: Gestionar la información de la prueba de urea de las judocas en el sistema.	
Tipo de tareas: Desarrollo	Puntos estimados: 0.4
Fecha de inicio: 22 de Abril del 2010	Fecha Fin: 23 de Abril del 2010
Programador(es) responsable(s): Omar Mora Pérez – Francisco Rodríguez González	
<p>Descripción</p>	

Esta tarea está referida a la inserción, modificación y consulta de los datos de la prueba de urea de las judocas en el sistema, así como la visualización gráfica de los mismos. Se definen los permisos de modificación y escritura de cada campo según se requiera. Se mostrará en pantalla la información correspondiente al pesaje diario.

Tabla 59. Crear interfaz visual de gestión de las historias clínicas de las judocas

Tarea	
Número de la tarea: 1	Número de HU: 9
Nombre de la tarea: Crear interfaz visual de gestión de las historias clínicas de las judocas	
Tipo de tareas: Diseño	Puntos estimados: 0.2
Fecha de inicio: 26 de Abril del 2010	Fecha Fin: 26 de Abril del 2010
Programador(es) responsable(s): Francisco Rodríguez González – Omar Mora Pérez	
Descripción	
<p>Esta tarea define el diseño de las pantallas que mostrarán los datos de la gestión (insertar, modificar y consultar) de la información referente a la historia clínica de las judocas al usuario. Se crean las tablas y formularios para mostrar la información correspondiente y para introducir los datos correspondientes en los casos necesarios.</p>	

Tabla 60. Gestionar la información de las historias clínicas de las judocas en el sistema

Tarea	
Número de la tarea: 2	Número de HU: 9
Nombre de la tarea: Gestionar la información de las historias clínicas de las judocas en el sistema.	
Tipo de tareas: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 27 de Abril del 2010	Fecha Fin: 28 de Abril del 2010

Programador(es) responsable(s): Omar Mora Pérez – Francisco Rodríguez González

Descripción

Esta tarea está referida a la inserción, modificación y consulta de los datos de las historias clínicas de las judocas en el sistema, así como la visualización gráfica de los mismos. Se definen los permisos de modificación y escritura de cada campo según se requiera. Se mostrará en pantalla la información correspondiente al pesaje diario.

Anexo 4.

Casos de prueba de las funcionalidades modificar, consultar y obtener reporte de la HU Test de proyecciones.

Tabla 61. Caso de Pruebas No. 2 de la HU Test de proyecciones.

Caso de prueba de Aceptación	
Código: HU4_P2	Historia de Usuario: 4
Nombre: Test de proyecciones	
Descripción: Prueba para la funcionalidad de modificar resultados de la prueba de proyecciones de las atletas.	
Condiciones de ejecución: El usuario debe estar autenticado. Debe de existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan planificado pruebas médicas.	
Entradas/Pasos de ejecución: Se selecciona el plan gráfico médico, el macrosistema, el mesosistema y el microsistema en los que se planificaron pruebas médicas y la fecha en que se realizó la prueba de proyecciones, luego se introducen los nuevos resultados al sistema.	
Resultado esperado: Se muestra un aviso de que se han modificado los datos correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 62. Caso de Pruebas No. 3 de la HU Test de proyecciones.

Caso de prueba de Aceptación	
Código: HU4_P3	Historia de Usuario: 4

Nombre: Test de proyecciones
Descripción: Prueba para la funcionalidad de consultar resultados de la prueba de proyecciones de las atletas.
Condiciones de ejecución: El usuario debe estar autenticado. Debe de existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan planificado pruebas médicas.
Entradas/Pasos de ejecución: Se selecciona el plan gráfico médico, el macrosistema, el mesosistema y el microsistema en los que se planificaron pruebas médicas y la fecha en que se realizó la prueba de proyecciones.
Resultado esperado: Se muestran los datos correspondientes de la prueba seleccionada.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 63. Caso de Pruebas No. 4 de la HU Test de proyecciones.

Caso de prueba de Aceptación	
Código: HU4_P4	Historia de Usuario: 4
Nombre: Test de proyecciones	
Descripción: Prueba para la funcionalidad de obtención del reporte de los resultados de la prueba de proyecciones de las atletas en períodos anteriores.	
Condiciones de ejecución: El usuario debe estar autenticado. Debe de existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan planificado pruebas médicas.	
Entradas/Pasos de ejecución:	

Se selecciona la atleta a la que se le desea mostrar el reporte de los resultados obtenidos en estas pruebas anteriormente.

Resultado esperado: Se muestran los datos correspondientes de las pruebas aplicadas anteriormente.

Evaluación de la prueba: Prueba satisfactoria.

Anexo 5.

Casos de prueba de las funcionalidades modificar, consultar y obtener reporte de la HU Laboratorio Clínico.

Tabla 64. Caso de Pruebas No. 2 de la HU Laboratorio clínico.

Caso de prueba de Aceptación	
Código: HU5_P2	Historia de Usuario: 5
Nombre: Laboratorio clínico	
Descripción: Prueba para la funcionalidad de modificar los resultados de las pruebas del laboratorio clínico de las atletas.	
Condiciones de ejecución: El usuario debe estar autenticado. Debe de existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan planificado pruebas médicas.	
Entradas/Pasos de ejecución: Se selecciona el plan gráfico médico, el macrosistema, el mesosistema y el microsistema en los que se planificaron pruebas médicas, luego se modifican los resultados de las pruebas de laboratorio clínico realizadas.	
Resultado esperado: Se muestra un aviso de que se han modificado los datos correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 65. Caso de Pruebas No. 3 de la HU Laboratorio Clínico.

Caso de prueba de Aceptación	
Código: HU5_P3	Historia de Usuario: 5

Nombre: Laboratorio clínico
Descripción: Prueba para la funcionalidad de consultar los resultados de las pruebas del laboratorio clínico de las atletas.
Condiciones de ejecución: El usuario debe estar autenticado. Debe de existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan planificado pruebas médicas.
Entradas/Pasos de ejecución: Se selecciona el plan gráfico médico, el macrosistema, el mesosistema y el microsistema en los que se planificaron pruebas médicas.
Resultado esperado: Se muestran los datos de las pruebas de laboratorio clínico realizadas en el período seleccionado.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 66. Caso de Pruebas No. 4 de la HU Laboratorio Clínico.

Caso de prueba de Aceptación	
Código: HU5_P4	Historia de Usuario: 5
Nombre: Laboratorio clínico.	
Descripción: Prueba para la funcionalidad de obtención del reporte de los resultados de las pruebas del laboratorio clínico de las atletas en períodos anteriores.	
Condiciones de ejecución: El usuario debe estar autenticado. Debe de existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan planificado pruebas médicas.	

Entradas/Pasos de ejecución:

Se selecciona la atleta a la que se le desea mostrar el reporte de los resultados obtenidos en esta prueba anteriormente.

Resultado esperado: Se muestran los resultados correspondientes de las pruebas aplicadas anteriormente.

Evaluación de la prueba: Prueba satisfactoria.

Anexo 6.

Casos de prueba de las funcionalidades modificar, consultar y obtener reporte de la HU Laboratorio de desarrollo físico.

Tabla 67. Caso de Pruebas No.2 de la HU Laboratorio de desarrollo físico.

Caso de prueba de Aceptación	
Código: HU6_P2	Historia de Usuario: 6
Nombre: Laboratorio de desarrollo físico	
Descripción: Prueba para la funcionalidad de modificar los resultados de las pruebas del laboratorio de desarrollo físico de las atletas en el sistema.	
Condiciones de ejecución: El usuario debe estar autenticado. Debe de existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan planificado pruebas médicas.	
Entradas/Pasos de ejecución: Se selecciona el plan gráfico médico, el macrosistema, el mesosistema y el microsistema en los que se planificaron pruebas médicas, luego se modifican los resultados de las pruebas de laboratorio de desarrollo físico ya desarrolladas.	
Resultado esperado: Se muestra un aviso de que se han modificado los datos correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 68. Caso de Pruebas No.3 de la HU Laboratorio de desarrollo físico.

Caso de prueba de Aceptación	
Código: HU6_P3	Historia de Usuario: 6

Nombre: Laboratorio de desarrollo físico
Descripción: Prueba para la funcionalidad de consultar los resultados de las pruebas del laboratorio de desarrollo físico de las atletas en el sistema.
Condiciones de ejecución: El usuario debe estar autenticado. Debe de existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan planificado pruebas médicas.
Entradas/Pasos de ejecución: Se selecciona el plan gráfico médico, el macrosistema, el mesosistema y el microsistema en los que se planificaron pruebas médicas.
Resultado esperado: Se muestran los resultados correspondientes a las pruebas de laboratorio cardiovascular desarrolladas.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 69. Caso de Pruebas No.4 de la HU Laboratorio de desarrollo físico.

Caso de prueba de Aceptación	
Código: HU6_P4	Historia de Usuario: 6
Nombre: Laboratorio de desarrollo físico.	
Descripción: Prueba para la funcionalidad de obtención del reporte de los resultados de las pruebas del laboratorio de desarrollo físico de las atletas en períodos anteriores.	
Condiciones de ejecución: El usuario debe estar autenticado. Debe de existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan planificado pruebas médicas.	

Entradas/Pasos de ejecución:

Se selecciona la atleta a la que se le desea mostrar el reporte de los resultados obtenidos en esta prueba anteriormente.

Resultado esperado: Se muestran los resultados correspondientes de las pruebas aplicadas anteriormente.

Evaluación de la prueba: Prueba satisfactoria.

Anexo 7.

Casos de prueba de las funcionalidades modificar, consultar y obtener reporte de la HU Laboratorio cardiovascular.

Tabla 70. Caso de Pruebas No.2 de la HU Laboratorio cardiovascular.

Caso de prueba de Aceptación	
Código: HU7_P2	Historia de Usuario: 7
Nombre: Laboratorio cardiovascular	
Descripción: Prueba para la funcionalidad de modificar los resultados de las pruebas del laboratorio cardiovascular de las atletas en el sistema.	
Condiciones de ejecución: El usuario debe estar autenticado. Debe de existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan planificado pruebas médicas.	
Entradas/Pasos de ejecución: Se selecciona el plan gráfico médico, el macrosistema, el mesosistema y el microsistema en los que se planificaron pruebas médicas, luego se modifican los resultados de las pruebas de laboratorio cardiovascular ya desarrolladas.	
Resultado esperado: Se muestra un aviso de que se han modificado los datos correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 71. Caso de Pruebas No.3 de la HU Laboratorio cardiovascular.

Caso de prueba de Aceptación	
Código: HU7_P3	Historia de Usuario: 7

Nombre: Laboratorio cardiovascular
Descripción: Prueba para la funcionalidad de consultar los resultados de las pruebas del laboratorio cardiovascular de las atletas en el sistema.
Condiciones de ejecución: El usuario debe estar autenticado. Debe de existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan planificado pruebas médicas.
Entradas/Pasos de ejecución: Se selecciona el plan gráfico médico, el macrosistema, el mesosistema y el microsistema en los que se planificaron pruebas médicas.
Resultado esperado: Se muestran los resultados correspondientes a las pruebas de laboratorio cardiovascular desarrolladas.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 72. Caso de Pruebas No.4 de la HU Laboratorio cardiovascular.

Caso de prueba de Aceptación	
Código: HU7_P4	Historia de Usuario: 7
Nombre: Laboratorio cardiovascular.	
Descripción: Prueba para la funcionalidad de obtención del reporte de los resultados de las pruebas del laboratorio cardiovascular de las atletas en períodos anteriores.	
Condiciones de ejecución: El usuario debe estar autenticado. Debe de existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan planificado pruebas médicas.	

Entradas/Pasos de ejecución:

Se selecciona la atleta a la que se le desea mostrar el reporte de los resultados obtenidos en esta prueba anteriormente.

Resultado esperado: Se muestran los resultados correspondientes de las pruebas aplicadas anteriormente.

Evaluación de la prueba: Prueba satisfactoria.

Anexo 8.

Casos de prueba de las funcionalidades modificar y consultar de la HU Historia clínica.

Tabla 73. Caso de Pruebas No.2 de la HU Historia clínica.

Caso de prueba de Aceptación	
Código: HU8_P2	Historia de Usuario: 8
Nombre: Historia clínica	
Descripción: Prueba para la funcionalidad de modificar los resultados de las historias clínicas confeccionadas a las atletas.	
Condiciones de ejecución: El usuario debe estar autenticado. Debe de existir una historia clínica confeccionada a la atleta seleccionada.	
Entradas/Pasos de ejecución: Se selecciona la atleta a la que se le desea modificar la historia clínica confeccionada con anterioridad y se identifican los padecimientos, enfermedades y lesiones a modificar.	
Resultado esperado: Se muestra un aviso de que se han modificado los datos correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 74. Caso de Pruebas No.3 de la HU Historia clínica.

Caso de prueba de Aceptación	
Código: HU8_P3	Historia de Usuario: 8
Nombre: Historia clínica	
Descripción: Prueba para la funcionalidad de consultar los resultados de las historias clínicas confeccionadas a las atletas.	

Condiciones de ejecución:

El usuario debe estar autenticado.

Debe de existir una historia clínica confeccionada a la atleta seleccionada.

Entradas/Pasos de ejecución:

Se selecciona la atleta a la que se le desea consultar la historia clínica confeccionada con anterioridad.

Resultado esperado: Se muestran los datos registrados de la atleta seleccionada en su historia clínica.

Evaluación de la prueba: Prueba satisfactoria.

Anexo 9.

Casos de prueba de las funcionalidades modificar, consultar y obtener reporte de la HU Prueba de Urea.

Tabla 75. Caso de Pruebas No.2 de la HU Prueba de urea.

Caso de prueba de Aceptación	
Código: HU9_P2	Historia de Usuario: 9
Nombre: Prueba de urea	
Descripción: Prueba para la funcionalidad de modificar los resultados de la prueba de urea realizadas a las atletas.	
Condiciones de ejecución: El usuario debe estar autenticado. Debe de existir una prueba de urea ya realizada.	
Entradas/Pasos de ejecución: Se selecciona el día en que desea cambiar los resultados de la prueba de urea ya realizada y se modifican los datos.	
Resultado esperado: Aparece un aviso de que se han modificado los datos correctamente.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 76. Caso de Pruebas No.3 de la HU Prueba de urea.

Caso de prueba de Aceptación	
Código: HU9_P3	Historia de Usuario: 10
Nombre: Prueba de urea	
Descripción: Prueba para la funcionalidad de consultar el resultado de las pruebas de urea de las	

atletas.
<p>Condiciones de ejecución:</p> <p>El usuario debe estar autenticado.</p> <p>Deben de existir pruebas de urea ya realizadas y registradas en el sistema.</p>
<p>Entradas/Pasos de ejecución:</p> <p>Se selecciona la fecha del día en que desean consultar los resultados ya existentes de cada atleta.</p>
<p>Resultado esperado: Se muestran los resultados de las pruebas de urea realizadas.</p>
<p>Evaluación de la prueba: Prueba satisfactoria.</p>

Tabla 77. Caso de Pruebas No.4 de la HU Prueba de urea.

Caso de prueba de Aceptación	
Código: HU9_P4	Historia de Usuario: 9
Nombre: Prueba de Urea.	
Descripción: Prueba para la funcionalidad de obtención del reporte de los resultados de las pruebas de urea de las atletas en períodos anteriores.	
<p>Condiciones de ejecución:</p> <p>El usuario debe estar autenticado.</p> <p>Debe de existir un plan gráfico médico, un macrosistema, un mesosistema y un microsistema en los que se hayan planificado pruebas médicas.</p>	
<p>Entradas/Pasos de ejecución:</p> <p>Se selecciona la atleta a la que se le desea mostrar el reporte de los resultados obtenidos en esta prueba anteriormente.</p>	
Resultado esperado: Se muestran los resultados correspondientes de las pruebas aplicadas anteriormente.	

Evaluación de la prueba: Prueba satisfactoria.