UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS

"Análisis, Diseño e Implementación del submódulo Atletismo en el módulo de Desarrollo Resultados Competitivos del Sistema SGID."

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Autor (es):
Martha Luisa Gala Rodríguez.
Darién Concepción Quintero.

Tutor: Ing. Herson Fernández Machado.

"Año del 50 aniversario del Triunfo de la Revolución"

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo 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 Junio del 2009.

Autores:

_____ _____

Firma del autor. Firma del autor.

Martha Luisa Gala Rodríguez Darién Concepción Quintero

_____

Firma del Tutor

Tutor: Ing.: Herson Fernández Machado

¿Por qué esta magnífica tecnología científica, que ahorra trabajo y nos hace la vida más fácil, nos aporta tan poca felicidad?
La repuesta es está, simplemente: Porque aún no hemos aprendido a usarla con tino.

Albert Einstein.

AGRADECIMIENTOS.

A mis padres, a mi abuelo, mi hermana, a mi familia en general, por que sin ellos no estaría aquí ahora siendo quien soy.

A todos los estudiantes de mi grupo por soportarme y por compartir conmigo durante estos 5 años.

A todos aquellos que de una forma u otra me ayudaron para llegar aquí y confiaron en mí.

A mis amigos, esos que siempre están aunque no me lo digan pues siempre tuvieron fé en mí, sobre todo a aquellos que me comprometieron a esforzarme para ser quien soy.

Muchas gracias a todos.

Martha Luisa Gala Rodríguez.

Mis agradecimientos van a mi mamá, a mi papá, a mi hermana, a mi abuelita que aunque ya no está, ha significado y significa mucho para mí y para mi carrera, porque por ellos y por su apollo estoy aquí.

A mis amigos Jose, Pipe, Nersida y a todos los que estuvieron conmigo en hasta ahora el momento mas difícil de mi vida, a todos los que se ocuparon de mi familia mientras estuve lejos de ella, a quienes me ayudaron y estuvieron pendientes de mi, de mi papá, de mi hermanita y de mi abuela

A todos mis compañeros de grupo, a los que llegaron al final y a los que no pudieron. Gracias por compartir juntos esta experiencia.

A todos mis amigos de la facultada los nuevos y a los de siempre, por apoyarme y confiar en mí.

A mi hermano Alexei, por coger tantos palos juntos y por toda su ayuda en estos 5 cortos años.

Graias por ayudarme y compartir conmigo todo este tiempo, gracias por confiar en mi hasta el final.

Darién Concepción Quintero.

DEDICATORIA.

Este trabajo diploma va dedicado especialmente a mis padres, por confiar en mi y darme el apoyo incondicionalmente en todo, guiandome siempre por el buen camino.

A mi comandante en jefe Fidel Castro por pensar en esta escuela poniendo en alto los intereses de los jóvenes que como yo quieren superarse y ser mejores cada día, dándome la oportunidad de formarme como profesional.

Martha Luisa Gala Rodríguez.

Dedico mi trabajo de diploma a mis padres y especialmente a mi abuelita que por estar aquí haciendo este trabajo no pude llegar a tiempo para que me viera antes de irse como estoy seguro que quería. Gracias por estar pendientes de mí y por el apoyo que recibí en todo momento, gracias por guiarme a hacer lo correcto.

A las FAR, por darme su ejemplo y educarme. A mi Comandante en Jefe por la oportunidad de vivir esta experiencia.

A todos lo que han puesto un pedacito de su corazón en mi y me dajaron poner un pedacito del mio en ellos.

Darién Concepción Quintero.

RESUMEN.

El presente trabajo diploma resuelve la polémica que existe en los eventos deportivos del atletismo en Cuba ya que mediante esta aplicación de escritorio se gestionan todas las estadísticas que son generadas por los deportistas en las diferentes modalidades del atletismo. Optimizando de forma general todo lo relacionado con las estadísticas de un evento determinado, logrando una mejor organización y belleza en cada evento, además minimiza el papeleo y agiliza el flujo de información ahorrando así tiempo y esfuerzo físico de los especialistas encargados de dicha tarea. Esta aplicación cuenta con una interfáz amigable y fácil de trabajar, pues no se necesita de mucho tiempo para aprender a interactuar con ella.

ABSTRACT.

The present work diploma solve the problem that he exists in the sporting events of athletics in Cuba right now than by means of this application of desk they try to obtain all of the statistics that are generated by sportsmen in the different formalities of athletics. Optimizing of general form all related matters with the statistics of a determined event, achieving a better organization and beauty in each event, besides he minimizes the paper work and he speeds up the flow of information saving thus time and I demand of entrusted specialists said task physically. This application has a friendly and easy interface to work, don't need a long time to learn to interact with her.

Índice:

Introducción.

Las distintas modalidades que componen la disciplina Atletismo o sea, las carreras, los saltos y lanzamientos, surgen con el hombre en la tierra, y están muy ligadas a la supervivencia de éste. También existe la certeza que en las civilizaciones más antiguas del continente europeo se competía en varios eventos, especialidades o modalidades del Atletismo, lo cual constituyó el deporte fundamental en las Olimpiadas de la Era Antigua. Los orígenes en Cuba también se pierden en la historia, pero de forma oficial marca un punto de partida la participación del país en los II Juegos Estivales de París 1900, donde el esgrimista Ramón Fonst se convirtió en el primer medallista dorado de la mayor de las Antillas.

El Atletismo moderno nació en el año 1866, las competencias se desarrollan en dos grandes áreas, los eventos de pista y de campo, además es un deporte de tiempos y marcas, donde un atleta rivaliza en última instancia, contra sus propios récords. La instalación para efectuar los enfrentamientos de Atletismo debe constar con las facilidades requeridas para las 44 especialidades, de ellas, 24 para hombres y 20 para mujeres. En 1912 surge la Federación Internacional de Atletismo, que gestiona este deporte y homologa las marcas. A partir de aquí las pruebas atléticas se repartieron en tres categorías: carreras, salto y lanzamiento.

En cualquier competencia de Atletismo laboran más de un centenar de jueces y árbitros y la organización de un evento es altamente compleja. Cuba es una potencia en el Atletismo y cuenta con figuras reconocidas en el mundo como Alberto Juantorena, Ana Fidelia Quirot, Javier Sotomayor, Iván Pedroso y los vallistas Anier García y Dayron Robles.

En Cuba se realizan cada año eventos deportivos en diversas disciplinas entre ellas el Atletismo la cual se desarrolla en todas sus modalidades. Dentro de estos eventos deportivos está la olimpiada del deporte cubano, así como maratones en conmemoración a distintas fechas celebradas, en el transcurso del tiempo. En estos eventos se recogen todos los datos correspondientes a los deportistas, así como los resultados individuales y por modalidades de forma manual. Esta información es desconocida para todo el personal ajeno a la competencia (espectadores), que aunque interesados en saberla no tienen forma de hacerlo, incluso en ocasiones el propio deportista desconoce el resultado que obtiene en cada salto, lanzamiento o carrera. Este proceso además de ser engorroso quita calidad al evento pues con frecuencia comienzan fuera del horario previsto con anterioridad, debido a todo este proceso de registrar y verificar el deportista en cada modalidad que participa, así como registrar sus resultados, pues como son muchos deportistas se deben buscar a todos y cada uno para registrar el dato del resultado obtenido.

Después de haber estudiado esta ***situación problemática*** se ha detectado un problema a resolver el cual se le dará solución mediante un Sistema de Gestión de Informatización del Deporte (SGID), pues en esta era de información digital no se deja fuera el deporte y comenzaremos desde la Universidad de las Ciencias Informáticas y se pretende extender el sistema a nivel nacional o internacional. El proyecto SGID posee diversas disciplinas de las cuales debe recopilar instantáneamente todos los datos estadísticos que se generen en un evento de la misma. Para ello se requiere del análisis, diseño e implementación de un submódulo, en este caso Atletismo, que permita introducir a la base de datos central todos los resultados que se están derivando de un desarrollo de la misma en un principio a nivel nacional y posteriormente como recomendación a nivel internacional. Esto permitirá primeramente que los datos sean recogidos y archivados en el instante que se generan, al igual que una mejor organización para la competencia a realizar y el evento en general. Además estará desarrollado sobre la filosofía del software libre como vía principal a la independencia y funcionalidad del software.

Para desarrollar este sistema se tiene presente un ***problema científico*** el cual se basa en: ¿Cómo gestionar la información de forma eficiente, segura y rápida en los eventos de Atletismo en la Universidad de las Ciencias Informáticas? Como consecuencia del problema anteriormente expuesto se plantea la siguiente **idea a defender:** con la confección de una aplicación que controle y archive la información referente a los eventos de atletismo en Cuba, se agilizaría y facilitaría el acceso a dicha información.

Este problema tiene como ***objeto de estudio***: la gestión de la información en el deporte. El objetivo delimita el ***campo de acción*** al definir la gestión de la información en el deporte de Atletismo. El ***objetivo principal*** que se persigue con el desarrollo de este submódulo es realizar el análisis, diseño e implementación del mismo, contribuyendo de esta forma a completar la aplicación que se desea crear integrando los módulos de las distintas disciplinas.

Objetivos específicos:

1. Desarrollar el análisis y diseño de un sistema informático que gestione todo el proceso relacionado con los Eventos de Atletismo.
2. Lograr un diseño de BD el cual permita el flujo y almacenamiento eficiente de la información a gestionar.
3. Implementar las funciones definidas para el sistema derivadas del análisis y diseño del mismo.

Para el desarrollo del submódulo de Atletismo se realizarán las siguientes tareas:

Tareas de la investigación:

1. Investigar las tendencias y tecnologías actuales para el desarrollo de Software de Gestión así como para la construcción de las Bases de Datos.

2. Investigar las metodologías de la ingeniería de software para el desarrollo del software de Gestión y las herramientas de modelado para la confección de los artefactos del sistema durante el desarrollo del software.
3. Seleccionar de las herramientas y tecnologías para el desarrollo de la aplicación.
4. Realizar el análisis y diseño de la aplicación.
5. Diseñar la Base de Datos.
6. Elaborar los documentos pertinentes para cada una de las fases en las que se este desarrollando el proyecto entiéndase fases de análisis y diseño, así como los datos recogidos en el levantamiento de requisitos.
7. Implementar los casos de uso del sistema y la base de datos así como la integración de los mismos para obtener el producto final.

Capítulo 1: Fundamentación teórica.

Introducción

En el presente capítulo se realiza un estudio de los principales sistemas que se han desarrollado, en Cuba, llegando finalmente a la toma de decisiones con respecto al sistema que se desea desarrollar. Se brinda una visión general de los aspectos relacionados con el Análisis, Diseño e Implementación del submódulo Atletismo en el módulo de Desarrollo Resultados Competitivos del Sistema de Gestión de Informatización del Deporte (SGID), explicando de forma clara y precisa los principales conceptos asociados al dominio de éste, proponiéndose un acercamiento a las tendencias y tecnologías actuales, fundamentando las razones que impulsaron la selección de la metodología, el lenguaje y herramientas que proponemos para el desarrollo satisfactorio del sistema.

Sistemas fuera de Cuba.

Olympic Organizer Deluxe 1.4

Este programa incluye plantillas predefinidas con las que podrás almacenar toda la información relacionada con los Juegos Olímpicos además de brindar la funcionalidad de modificar las mismas a gusto propio, los atletas que consigan medallas, los diarios de los campeones, enlaces a páginas web relacionadas con los juegos, así como una extensa base de datos relacionada con los medallistas de los Juegos Olímpicos. El programa permite realizar impresiones de todos los datos almacenados dando la posibilidad de personalizar el modo de impresión. Permite al cliente definir la forma de organización de los datos, que quiere mostrar dándole control total sobre la base de datos. Organiza y exhibe los datos eficazmente, a fin de que la información sea fácilmente accesible de diversos modos. Permite una búsqueda flexible mediante filtros de selección. Es un software propietario, compatible solo con Windows 95, 98, 2000, NT.

Sistemas en Cuba

En cuba existen sitios web como http://www.inder.cu que muestran datos he información sobre los distintos eventos que se realizan en el país, así como los datos relacionados con deportistas cubanos participantes en cualquiera de esos eventos, este sitio permite además el acceso a otros sitios deportivos provinciales. También esta http://www.cubadeportes.cu que brinda la información correspondiente a los distintos

servicios que se brindan a los participantes de estos eventos como son los Servicios Técnicos Especializados, Atención al Viajero, pero ninguno de ellos brinda la posibilidad de gestionar los datos relacionados con las estadísticas de un evento determinado de Atletismo. Aunque es valido resaltar que ya se han desarrollado sistemas que gestionan información generada durante los juegos de beisbol en la serie nacional y que además se están implementando otros software dentro de la Universidad de la Ciencias Informáticas (UCI) que tienen mucho que ver con este deporte en particular.

En la Universidad de las Ciencias Informáticas existe un software que muestra los resultados de cada evento deportivo de forma general o sea de cada disciplina por separado y después de haber ocurrido este, el mismo es conocido como el sitio de los Juegos Inter-Facultades que lleva el nombre de los juegos que se determina ese mismo año; el cual no satisface la necesidad de brindar información instantáneamente sino que es actualizado con posterioridad, lo que provoca que el espectador esté de cierta forma desinformado con lo que está ocurriendo en ese instante de la competencia.

Principales conceptos para el dominio del problema.

Atletismo**,** término que denota un conjunto de pruebas celebradas como competiciones entre individuos o equipos —normalmente amateurs— en reuniones en pista cubierta o estadios al aire libre. Las categorías básicas del deporte combinan pruebas de carreras y marchas con lanzamientos y saltos. Las carreras, que constituyen la mayor parte de las pruebas de atletismo, varían desde los 50 m lisos en pista cubierta, hasta la carrera de maratón, que cubre 41 km. 947 m. En Estados Unidos y Gran Bretaña las distancias se expresaban en millas, pero desde 1976, para récords oficiales, sólo se reconocen distancias métricas (excepto para la carrera de la milla). En Europa y en los Juegos Olímpicos las distancias se han expresado siempre en metros. En este artículo se usa el sistema métrico para expresar las distancias.

He aquí las diferentes especialidades o modalidades de las que consta el ATLETISMO:

Carreras de velocidad

Las carreras más cortas son las de velocidad, que en pista cubierta se corren sobre distancias de 50 y 60 m y al aire libre sobre 100, 200 y 400 m. En este tipo de carreras, el atleta se agacha en la línea de salida, y al sonido del disparo de un juez de salida se lanza a la pista y corre a la máxima velocidad hacia la línea de meta, siendo fundamental una salida rápida. Los corredores alcanzan la tracción situando los pies contra unos bloques especiales de metal o plástico llamados estribos, diseñados especialmente para sujetar al corredor y que están colocados justo detrás de la línea de salida. Las características principales de un estilo eficiente para carreras de velocidad comprenden una buena elevación de rodillas, movimientos libres de los brazos y un ángulo de penetración del cuerpo de unos 25 grados.

Los corredores pueden usar diversas estrategias durante las carreras. En una carrera de 400 m, por ejemplo, el corredor puede correr a la velocidad máxima durante 200 m, luego relajarse un poco otros 150 m, para finalizar de nuevo con otro tope de velocidad. Algunos corredores prefieren correr 200 o 300 m a la máxima velocidad y luego intentar resistir el resto de la carrera. Cuando el corredor aminora la marcha, lo hace para

conservar energía, que utilizará en el momento en que efectúe de nuevo un esfuerzo máximo.

Vallas

Las pruebas de vallas son carreras de velocidad en las que los competidores deben superar una serie de diez barreras de madera y metal (o plástico y metal) llamadas vallas. Las carreras de vallas al aire libre más populares, para hombres y mujeres, son los 110 m vallas, que se corren con las denominadas vallas altas; los 400 m (con vallas intermedias) y los 200 m, con vallas bajas. En los campeonatos nacionales en pista cubierta se suelen correr los 60 m vallas. Las vallas altas miden 107 cm de altura, las intermedias 91 y las bajas 76.

En todas las distancias hasta los 110 m inclusive, la primera valla está a 13,72 m de la línea de salida y el resto de las vallas están separadas 9,14 m; la distancia desde la última valla hasta la meta es 14,02 m. En distancias superiores a 110 m pero que no exceden de 200, la primera valla está a 18 m de la salida y el resto están separadas 18 m una de la otra. En los 400 m, la primera valla está a 45 m y el resto están separadas 35 m, quedando 43 m desde la última valla hasta la meta.

En la prueba femenina de 100 m vallas, la primera está a 13 m de la salida y la separación entre ellas es de 8,5 m, quedando 10,5 m desde la última valla hasta la meta.

Una buena forma de saltar vallas consiste en saltar desde lejos y salvar las barreras suavemente sin romper el ritmo de la zancada. La primera pierna que pasa la valla vuelve a la pista de forma rápida; la otra pierna, mientras tanto, supera la valla casi en ángulo recto con respecto al cuerpo. Gran velocidad de carrera, flexibilidad y una gran coordinación, son elementos importantes para tener éxito.

Carreras de relevos

Las carreras de relevos son pruebas para equipos de cuatro componentes en las que un corredor recorre una distancia determinada, luego pasa al siguiente corredor un tubo rígido llamado testigo y así sucesivamente hasta que se completa la distancia de la carrera. El pase del testigo se debe realizar dentro de una zona determinada de 18 m de largo. En las carreras de 400 y 800 m relevos, el testigo pasa del corredor que lo entrega al que lo recibe cuando éste último ya ha comenzado a correr hacia adelante, continuando el receptor la carrera. En carreras más largas, debido a la fatiga acumulada, el corredor que recibe el testigo muchas veces mira hacia atrás para recogerlo. En condiciones ideales, tanto el corredor que entrega el testigo como el que lo recibe, deben ir a la máxima velocidad y separados unos dos metros al hacer el cambio. En estas pruebas, los miembros de los equipos que intervienen corren por una zona de la pista; para completar cada relevo el corredor debe entrar en la llamada zona de traspaso, que permite al receptor del testigo iniciar su carrera.

Carreras de media distancia

Las carreras que cubren entre 600 y 3.000 m se conocen como carreras de media distancia. Las más populares son las de 800 m, 1.500 m y 3.000 m. Aunque no es una modalidad olímpica ni se disputa en los campeonatos del mundo, otra carrera que se

mantiene en el calendario atlético es la carrera de la milla, de las que existen algunas famosas por el nombre de la ciudad donde se celebran. La prueba es muy popular y los corredores de élite la cubren con regularidad por debajo de los 3 m 50 seg. El primer corredor que logró bajar de cuatro minutos fue el inglés Roger Bannister, que en 1954 lo hizo en 3 m 59,4 segundos.

Los competidores en carreras de media distancia deben regular su velocidad cuidadosamente para evitar quedarse exhaustos; algunos cambian de ritmo varias veces durante la carrera mientras que otros mantienen el mismo toda la prueba. El corredor finlandés Paavo Nurmi, que ganó una medalla de oro en la prueba de 1.500 m, en los Juegos Olímpicos de 1924, llevaba un cronómetro durante las carreras para comprobar su marcha. La forma de correr más apropiada para las carreras de media distancia difiere de la utilizada en las carreras de velocidad. La acción de rodillas es mucho menos pronunciada, la zancada es más corta y el ángulo hacia adelante del cuerpo es menos acusado.

Carreras de larga distancia

Las carreras por encima de 3.000 m se consideran pruebas de larga distancia. Estas carreras son muy populares en Europa, donde se celebran con frecuencia carreras de 5.000 y 10.000 metros. El estilo utilizado por los campeones de la distancia evita cualquier exceso en los movimientos; la acción de rodillas es ligera, los movimientos de los brazos se reducen al mínimo y las zancadas son más cortas que las de las carreras de velocidad o media distancia.

Entre las carreras más agotadoras de larga distancia están las de campo a través (o cross) y maratón. A diferencia de otras carreras de larga distancia, que se corren sobre pistas de composición variada, las de campo a través se realizan sobre un terreno tosco y natural. Debido a la variedad de condiciones y lugares no suele haber récords en éste tipo de carreras. Las carreras de maratón se corren normalmente sobre carreteras pavimentadas. Los corredores de ambas pruebas deben aprender a ascender colinas con zancadas cortas y eficientes y a descender con rapidez sin agitarse ni descontrolar el paso. Es esencial un paso uniforme y mantenido.

Maratón

La prueba más larga de las competiciones de campo y pista (42,195 km), y la última de los Juegos Olímpicos de Verano. La maratón ha llegado a ser muy popular y las carreras más importantes —como las celebradas cada año en Londres, Boston, Chicago y Nueva York— pueden atraer a más de 20.000 corredores y cientos de miles de espectadores.

La distancia se escogió para replicar la realizada por un soldado griego desde la ciudad de Maratón hasta la de Atenas en el año 490 a.C. para llevar la noticia de la victoria de los griegos sobre los persas; la distancia actual de Maratón a Atenas es menor de 40 km. La distancia moderna se estableció en los Juegos Olímpicos de Londres (1908) y representa la distancia desde el castillo real de Windsor hasta el estadio de White City en el oeste de la ciudad. Desde el inicio de las olimpiadas modernas en 1896 hasta 1984 sólo corrieron hombres en la prueba. No hay récords masculinos ni femeninos de la prueba porque cada recorrido de maratón es diferente. Los mejores tiempos masculinos están por debajo de 2h 7m y los mejores femeninos algo superior a 2 h 20 m.

Corredores que alcanzaron el éxito en esta prueba fueron el etíope Abebe Bikila, que ganó la carrera de maratón en las olimpiadas de 1960 y 1964, y el alemán Waldemar Cierpinski, que ganó la medalla de oro en los Juegos Olímpicos de 1976 y 1980. Las actuaciones de la noruega Grete Waitz ayudaron a levantar la popularidad de la prueba entre las mujeres a partir de la década de 1980; Waitz ganó la maratón de Nueva York nueve veces, desde 1978 hasta 1988, así como la maratón de Londres en dos ocasiones. Ingrid Kristiansen, otra noruega, ganó la maratón de Londres en cuatro ocasiones durante la década de 1980 y en 1985 volvió a hacerlo con una marca personal que hoy en día, once años después, sigue siendo el récord del mundo. En los Juegos Olímpicos de Atlanta (1996) ganó el surafricano J. Thugwane y en la prueba femenina F. Roba de Etiopía.

Marcha

Las pruebas de marcha se corren normalmente sobre distancias que oscilan entre 1.500 m y 50 km y son especialmente populares en Europa y Estados Unidos.

La regla principal de este tipo de carreras es que el talón del pie delantero debe permanecer en contacto con el suelo hasta que la puntera del pie de atrás deje de hacer contacto con el mismo. La regla está diseñada para evitar que corran los participantes.

Salto de altura

El objetivo en el salto de altura es pasar sobre una barra horizontal que se encuentra suspendida entre dos soportes verticales separados unos cuatro metros. El participante tiene derecho a tres intentos para superar cada altura. La mayoría de los saltadores de hoy en día usan el estilo de batida denominado *fosbury*, denominada así en homenaje a su inventor, el saltador estadounidense Dick Fosbury, quien la usó por primera vez en los Juegos Olímpicos de 1968. Para ejecutar el salto, los saltadores se aproximan a la barra casi de frente, se giran en el despegue, alcanzan la barra con la cabeza por delante, superándola de espaldas y caen en la colchoneta con sus hombros.

Salto de pértiga

En salto de pértiga, el atleta intenta superar una barra transversal situada a gran altura con la ayuda de una pértiga flexible, normalmente de 4 a 5 m de longitud y que suele ser de fibra de vidrio desde que reemplazara al bambú y al metal en la década de 1960. El saltador agarra la pértiga unos centímetros antes del final de la misma, corre por la pista hacia donde se encuentra la barra, clava la punta de la pértiga en un pequeño foso o agujero que está situado inmediatamente antes de donde se encuentra la proyección de la barra y salta hacia arriba impulsándose con la pértiga, cruza el listón con los pies por delante y luego cae en la colchoneta.

Los participantes tienen tres intentos para cada altura que va aumentando en 5 cm cada vez. Tres fallos en una altura determinada descalifican al saltador. Al competidor se le concede entonces como marca personal la última altura superada durante la prueba. Los fallos son: tirar el listón, pasar por un lado, pasar por debajo, colocar la pértiga más lejos de donde se encuentra el cajón de tomar impulso, cambiar las manos en el agarre de la pértiga y mover la mano de arriba durante el salto. Los saltos se miden perpendicularmente desde la parte de arriba de la barra hasta el suelo. En 1988 el

ucraniano Sergei Bubka, considerado el mejor saltador de pértiga de la historia, se convirtió en el primer atleta que superó los 6 m de altura. El salto de pértiga requiere una buena velocidad de carrera, músculos fuertes en la espalda y una gran habilidad gimnástica.

Salto de longitud

En el salto de longitud el competidor corre por una pista y salta desde una plataforma intentando cubrir la máxima distancia posible. En pleno salto, el atleta tira de los pies hacia delante del cuerpo para ayudar a conseguir más distancia. Los competidores hacen tres saltos y los siete mejores pasan a la siguiente ronda de otros tres saltos. Un salto se mide en línea recta desde el borde frontal de la plataforma de despegue hasta la marca más cercana a la citada plataforma hecha por cualquier parte del cuerpo del atleta al contactar con la tierra. Los atletas se clasifican basándose en sus saltos más largos. El salto de longitud requiere piernas fuertes, buenos músculos abdominales, velocidad de carrera y un brinco potente.

Triple salto

El objetivo en el triple salto es cubrir la máxima distancia posible en una serie de tres saltos entrelazados. En la primera fase de la secuencia, el saltador corre por la pista y salta desde una plataforma de lanzamiento cayendo en tierra con un pie, volviendo a impulsarse hacia adelante y cayendo con el pie opuesto, impulsándose de nuevo hacia arriba y adelante cayendo esta vez con ambos pies en la superficie preparada de tierra, de una forma similar a como lo hacen en el salto de longitud.

Lanzamiento de peso

El objetivo en el lanzamiento de peso es propulsar una sólida bola de metal a través del aire a la máxima distancia. El peso de la bola en hombres es de 7,26 kg y en mujeres 4 kg. La acción en el lanzamiento está circunscrita a un círculo de 2,1 m de diámetro.

En la primera fase de la prueba, el atleta sujeta el peso con los dedos de la mano de lanzar contra su hombro, poniendo la bola debajo de la barbilla. El competidor entonces salta o brinca dentro del círculo en una postura semi-agachada, adquiriendo velocidad. Al alcanzar el lado opuesto del círculo, estira el brazo de lanzar repentinamente y empuja el peso hacia el aire en la dirección adecuada. El peso se empuja, no se lanza.

El empuje se hace desde el hombro con un sólo brazo y no se puede llevar el peso detrás del hombro. Cada competidor tiene derecho a tres lanzamientos y los siete mejores pasan a la siguiente ronda de otros tres lanzamientos por competidor. Las medidas se efectúan desde el borde interno de la circunferencia del área de lanzamiento hasta el punto de impacto. Los competidores se clasifican de acuerdo a su mejor lanzamiento. Si el lanzador se sale del círculo, el lanzamiento es descalificado.

Lanzamiento de disco

El disco es un plato con el borde y el centro de metal que se lanza desde un círculo que tiene un diámetro de 2,5 m. En la competición masculina, el disco mide entre 219 y 221 mm de diámetro y de 44 a 46 mm de ancho; pesa 2 kg; en la femenina, mide entre

180 y 182 mm de diámetro y de 37 a 39 mm de ancho y pesa 1 kg. El atleta sujeta el disco plano contra los dedos y el antebrazo del lado del lanzamiento, luego gira sobre sí mismo rápidamente y lanza el disco al aire con una extensión del brazo.

El círculo está marcado por fuera con una tira metálica o pintura blanca. Dos líneas rectas se extienden hacia el exterior, desde el centro del círculo, formando un ángulo de 90º y para que los lanzamientos sean considerados válidos deben caer entre estas dos líneas. Una vez que los atletas entran en el círculo y comienzan el lanzamiento no pueden tocar el terreno de fuera del mismo hasta que el disco caiga en el suelo.

Los lanzamientos se miden desde el punto de impacto hasta la circunferencia interna del círculo en línea recta. Cada competidor hace tres lanzamientos, después de los cuales, los siete mejores pasan a la siguiente ronda de otros tres lanzamientos. Todos los lanzamientos cuentan y los atletas se clasifican con arreglo a sus mejores marcas.

Lanzamiento de martillo

Los lanzadores de martillo compiten lanzando una bola pesada adosada a un alambre metálico con un asidero en el extremo. La bola, el alambre y el asa juntos pesan 7,26 kg y forman una unidad de una longitud máxima de 1,2 m. La acción tiene lugar en un círculo de 2,1 m de diámetro. Agarrando el asa con las dos manos y manteniendo quietos los pies, el atleta hace girar la bola en un círculo que pasa por encima y por debajo de su cabeza, hasta la altura de las rodillas. Cuando el martillo alcanza velocidad, el lanzador gira sobre sí mismo dos o tres veces para acelerar aún más la bola del martillo y luego la suelta hacia arriba y hacia delante en un ángulo de 45º. Si el martillo no cae en el terreno dentro de un arco de 90º el lanzamiento no es válido. Cada lanzador hace tres intentos, pasando los siete mejores a la siguiente tanda de otros tres lanzamientos. Se comete una falta o violación de las reglas cuando cualquier parte del lanzador o del martillo toca fuera del círculo antes de que se haya completado el lanzamiento, es decir, que el martillo se haya parado en el suelo después de caer en el mismo. Los lanzadores de martillo suelen ser altos y con fuertes músculos, pero el éxito en los lanzamientos requiere también habilidad y coordinación. En las competiciones en pista cubierta se usa un martillo de 15,9 kg de peso.

Lanzamiento de jabalina

La jabalina es un venablo alargado con la punta metálica que tiene una longitud mínima de 260 cm para los hombres y 220 cm para las mujeres, y un peso mínimo de 800 g para los hombres y 600 g para las mujeres. Tiene un asidero de cordel de unos 15 cm de largo que se encuentra aproximadamente en el centro de gravedad.

Dos líneas paralelas separadas 4 m marcan la pista de lanzamiento de jabalina. La línea de lanzamiento tiene 7 cm de anchura y se encuentra alojada en el suelo tocando los extremos frontales de las líneas de marca de la pista. El centro de este pasillo está equidistante entre las líneas de marca de pista. Desde este punto central se extienden dos líneas más allá de la línea de lanzamiento hasta una distancia de 90 m. Todos los lanzamientos deben caer entre estas dos líneas.

Los lanzamientos se miden desde el punto de impacto hasta el punto central, pero sólo la distancia desde el lado interno del arco es válida. Los lanzadores deben permanecer en

la pista y no tocar o pasar la línea de lanzamiento. La jabalina debe caer primero con la punta. Los participantes hacen tres lanzamientos y los siete mejores pasan a la siguiente tanda de otros tres lanzamientos. Las clasificaciones se basan en el mejor lanzamiento realizado por cada competidor.

En el inicio de la acción, los competidores agarran la jabalina cerca de su centro de gravedad y corren velozmente hacia una línea de marca; al llegar a ella, se giran hacia un lado de su cuerpo, echan hacia atrás la jabalina y preparan el lanzamiento. Entre tanto, para mantener la velocidad durante la carrera mientras se echan hacia atrás para lanzar, dan un paso lateral rápido. Al llegar a la línea de marca, pivotan hacia adelante abruptamente y lanzan la jabalina al aire. El lanzamiento se invalida si cruzan la línea de lanzamiento o la jabalina no cae primero con la punta.

Metodología.

El proceso de desarrollo del software es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos se transforman en diseño y el diseño implementado en el código, este último es probado, documentado y certificado para su uso operativo. En otras palabras define quién hace qué artefactos, cómo y cuándo los realiza.

eXtreme Programming o Programación Extrema (XP)

Extreme Programming, se basa en el trabajo orientado directamente al objetivo, basándose para esto en las relaciones interpersonales y en la velocidad de reacción para la implementación y para los cambios que puedan surgir durante el desarrollo del proceso. Esto se logra, minimizando el riesgo de fallo del proceso manteniendo dentro del equipo a un representante "competente" del cliente, este representante es quién responderá a todas las preguntas y dudas que surjan por parte del equipo de desarrollo durante el proceso, de forma que no se retrase la toma de decisiones.

El objetivo de XP es generar versiones de la aplicación tan pequeñas como sea posible, pero que proporcionen un valor adicional, desde el punto de vista del negocio.

XP define un "diseño tan simple como sea posible" como aquél que:

- Pasa todos los test.
- No contiene código duplicado.
- Deja clara la intención de los programadores (enfatiza el qué, no el cómo) en cada línea de código.
- Contiene el menor número posible de clases y métodos.

Principales artefactos que se generan en XP

XP se basa en **User Stories (historias de uso),** estas historias las escribe el cliente o su representante dentro del equipo y describen los escenarios claves del funcionamiento del software sin entrar en muchos detalles. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario. Cuando llega

la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia.

Release planning (Plan de entregas): Después de tener ya definidas las historias de usuario es necesario crear un plan de publicaciones, en inglés "Release plan", donde se indiquen las historias de usuario que se crearán para cada versión del programa y las fechas en las que se publicarán estas versiones. Un "Release plan" es una planificación donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario, la prioridad con la que serán implementadas y las historias que serán implementadas en cada versión del programa. Después de un "Release plan" tienen que estar claros estos cuatro factores: los objetivos que se deben cumplir (que son principalmente las historias que se deben desarrollar en cada versión), el tiempo que tardarán en desarrollarse y publicarse las versiones del programa, el número de personas que trabajarán en el desarrollo y cómo se evaluará la calidad del trabajo realizado.

Iteration Plan (Plan de Iteraciones): Todo proyecto que siga la metodología X.P. se ha de dividir en iteraciones de aproximadamente de 1 a 3 semanas de duración. Al comienzo de cada iteración los clientes deben seleccionar las historias de usuario definidas en el "Release planning" que serán implementadas. También se seleccionan las historias de usuario que no pasaron el test de aceptación que se realizó al terminar la iteración anterior. Estas historias de usuario son divididas en tareas de entre 1 y 3 días de duración que se asignarán a los programadores.

Las características fundamentales del método son:

- **Desarrollo iterativo e incremental**: pequeñas mejoras, unas tras otras.
- **Pruebas unitarias continuas**, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- **Programación en parejas**: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera el código es revisado y discutido mientras se escribe es más importante que la posible pérdida de productividad inmediata.
- Frecuente **integración del equipo de programación con el cliente** o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- **Corrección de todos los errores** antes de añadir una nueva funcionalidad. Hacer entregas frecuentes. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- **Refactorización del código**, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- **Propiedad del código compartida**: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto.
- **Simplicidad en el código**: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La

programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Mientras más simple es el sistema, menos tendrá que comunicar sobre este, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores.

Feedback

La agilidad se define (entre otras cosas) por la capacidad de respuesta ante los cambios que se van haciendo necesarios a lo largo del camino. Por este motivo uno de los valores que nos hace más ágiles es el continuo seguimiento o feedback que recibimos a la hora de desarrollar en un entorno de desarrollo ágil. Este feedback se toma del cliente, de los miembros del equipo, en cuestión de todo el entorno en el que se mueve un equipo de desarrollo ágil. La Programación Extrema asume que la planificación nunca será perfecta, y que variará en función de cómo varíen las necesidades del negocio. Por tanto, el valor real reside en obtener rápidamente un plan inicial, y contar con mecanismos de feedback que permitan conocer con precisión dónde estamos. La obtención de feedback que permita llevar a cabo estimaciones precisas es fundamental. Se hacen estimaciones para cada historia, de modo que en cuanto se comienzan a tener datos históricos, éstos se utilizan para hacer que las siguientes estimaciones sean más precisas.

Ciclo de vida de XP

1ª Fase: Planificación del proyecto.

El primer paso de cualquier proyecto que siga la metodología X.P es definir las historias de usuario con el cliente. Después de tenerlas ya definidas es necesario crear un plan de publicaciones, donde se indiquen cuales son las que se crearán para cada versión del programa y las fechas en las que se publicarán estas versiones. Luego se divide en iteraciones y al comienzo de cada una, el cliente deben seleccionar las historias de usuario definidas en el plan de entregas que serán implementadas. Se seleccionan además las historias del usuario que no pasaron el test de aceptación que se realizo al terminar la iteración anterior. La programación debe ser en parejas pues incrementa la productividad y la calidad del software desarrollado. Esto implica que estén dos programadores trabajando juntos mientras uno codifica haciendo hincapiés en la calidad de la función, el otro analiza si esa función es adecuada y esta bien diseñada. Se deben realizar reuniones diarias para que los desarrolladores expongan sus problemas, soluciones e ideas de forma conjunta.

2ª Fase: Diseño.

La metodología XP sugiere diseños simples y sencillos, para que sean fácilmente entendibles, y su implementación más fácil. Se debe crear un glosario de términos y una correcta especificación de los nombres de los métodos y clases, que ayudará a la mejor comprensión para el diseño y facilitará sus posteriores ampliaciones y la reutilización del código. Como todo proyecto esta sujeto a riesgos XP sugiere utilizar una pareja de

desarrolladores para que investiguen y reduzcan el máximo riesgo que supone el problema. Este equipo tiene la tarea de no implementar ninguna funcionalidad extra con el pretexto de que en algún momento pueda servir, ya que sólo se utilizan el 10% de estas funcionalidades, además al implementarlas se desperdicia tiempo y recursos. Es necesario refactorizar ya que al usar códigos ya implementados que contienen funcionalidades que no serán usadas puede generar código completamente inestable y mal diseñado.

3ª Fase: Codificación.

A la hora de codificar una historia de usuario la presencia del cliente es aún más necesaria, ya que son los principales responsables de su creación y negocian los tiempos en los que serán implementadas. Se deben crear test que prueben el funcionamiento de los distintos códigos a implementar, ya que esto ayuda a saber lo que realmente tiene que hacer el código, además de que te da la certeza de que una vez implementado pasara dichos test sin problemas ya que fueron diseñados con ese objetivo. La codificación debe hacerse ateniendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad. Se usan repositorios donde las parejas de programadores publican cada pocas horas sus códigos implementados para el uso del equipo de desarrolladores, de esta forma los demás desarrolladores pueden acceder a estos códigos, teniendo la posibilidad de modificarlos o ampliar una clase o un método de otro programador y subirlo al repositorio. El permitir al resto de los programadores modificar códigos que no son suyos no supone ningún riesgo ya que para que un código pueda ser publicado en el repositorio tiene que pasar los test de funcionamiento definidos para el mismo. Los test se obtendrán de la especificación de requisitos ya que en ella se especifican las pruebas que deben pasar las distintas funcionalidades del programa, procurando codificar pensando en las pruebas que debe pasar cada funcionalidad.

La optimización del código siempre se debe dejar para el final. Hay que hacer que funcione y que sea correcto, más tarde se puede optimizar.

4ª Fase: Pruebas.

Uno de los pilares de la metodología X.P es el uso de test para comprobar el funcionamiento de los códigos que se vallan implementando.

La ejecución automatizada de un test es un elemento clave de XP. Existen tanto test internos (o de unidad), para garantizar que el mismo es correcto, como test de aceptación, para garantizar que el código hace lo que debe hacer. El cliente es el responsable de definir los de aceptación, no necesariamente de implementarlos. Él es la persona mejor calificada para decidir cuál es la funcionalidad más valiosa. Desde el punto de vista de XP, si no hay test, las cosas sólo funcionan en apariencia. Aún más, si un test no está automatizado, no se le puede considerar como tal; su objetivo no es corregir los errores, sino prevenirlos ya que se escriben antes que el código a testear, aportando así un gran valor adicional, pues fuerza a los desarrolladores a pensar como se va a usar el código. Elaborar los test exige pensar cuales son los problemas mas graves que se puedan presentar y cuales son los puntos dudosos evitando así muchos problemas y dudas que pueden aparecer en la marcha.

Al ser las distintas funcionalidades de nuestra aplicación no demasiado extensas, no se harán test que analicen partes de las mismas, sino que se realizarán para las funcionalidades generales que debe cumplir el programa especificado en la descripción de requisitos.

Rational Unified Process (RUP)

El Proceso Unificado de Desarrollo de Software, es un proceso que captura las mejores prácticas del conocimiento de líderes en ingeniería de Software y proporciona a los equipos de desarrollo guías, estándares y recomendaciones para la construcción de software de alta calidad. Es una forma disciplinada de asignar tareas y responsabilidades en una empresa de desarrollo (quién hace qué, cuándo y cómo). RUP es una guía de cómo usar Lenguaje de Modelado Unificado (UML) de la forma más efectiva. RUP sigue un modelo iterativo que aborda primero las tareas más riesgosas logrando reducir los riesgos del proyecto y tener un subsistema ejecutable tempranamente.
Tiene como principal objetivo asegurar la producción de software de calidad dentro de plazos y presupuestos predecibles. Dirigido por casos de uso, centrado en la arquitectura, iterativo (mini-proyectos) e incremental (versiones).

Dirigido por casos de uso: Basándose en los casos de uso, los desarrolladores crean una serie de modelos de diseño e implementación que los llevan a cabo. Además, estos modelos se validan para que sean conformes a los casos de uso. Finalmente, los casos de uso también sirven para realizar las pruebas sobre los componentes desarrollados.

Centrado en la arquitectura: En la arquitectura de la construcción, antes de construir un edificio éste se contempla desde varios puntos de vista: estructura, conducciones eléctricas, fontanería, etc. Cada uno de estos aspectos está representado por un gráfico con su notación correspondiente. Siguiendo este ejemplo, el concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema.

Iterativo e incremental: Todo sistema informático complejo supone un gran esfuerzo que puede durar desde varios meses hasta años. Por lo tanto, lo más práctico es dividir un proyecto en varias fases. Actualmente se suele hablar de ciclos de vida en los que se realizan varios recorridos por todas las fases. Cada recorrido por las fases se denomina iteración, en la que se realizan varios flujos de trabajo. Además, cada iteración parte de la anterior incrementado o revisando la funcionalidad implementada. Se suele denominar proceso.

Este es uno de los procesos más generales que existe, está enfocado a cualquier tipo de proyecto así no sea de software, se basa en la documentación generada en cada uno de sus cuatro fases:

1. Inicio (puesta en marchar)

2. Elaboración (definición, análisis y diseño)

3. Construcción (implementación)

4. Transición (fin del proyecto y puesta en producción) en las cuales se ejecutarán varias iteraciones (según el tamaño del proyecto).

RUP se basa en casos de uso para describir lo que se tiene y lo que se espera del software, está muy orientado a la arquitectura del sistema a implementarse, documentándose de la mejor manera, basándose en UML.

Para poder usar RUP antes hay que adaptarlo a las características de la empresa, y medir de manera exacta el tiempo, costos y todos los demás recursos involucrados en el proceso.

Feature Driven Development (FDD)

Este proceso se considera como punto medio entre los procesos pesados y ágiles, aunque en la práctica es más similar a este último. Pensado para proyectos relativamente cortos, al igual que los anteriores también esta basado en iteraciones que producen un software funcional que puede ser visto, probado y monitorizado por el cliente.

Estas iteraciones son decididas en base a las funcionalidades que el software debe tener, funcionalidades definidas por el cliente, este proceso está dividido en cinco fases:

1. Desarrollo de un Modelo General.

2. Construcción de la Lista de Funcionalidades.

3. Plan de releases basadas en las funcionalidades a implementar.

4. Diseñar en base a las funcionalidades definidas.

5. Implementar en base a las mismas funcionalidades.

Aquí en el equipo de trabajo sí existen jerarquías, siempre debe haber un jefe de proyecto, y aunque es un proceso considerado ligero también incluye documentación (la mínima necesaria para que algún nuevo integrante pueda entender el desarrollo de inmediato).

Comparando los procesos.

Esto es más que necesario, pues es lo que nos ayudará a elegir el mejor proceso a implementar para el desarrollo de un software, aquí lo necesario es saber con que recursos contamos (tiempo, dinero, personal) pues depende mucho de estos recursos el poder saber elegir un proceso y claro está debemos tener muy en claro el alcance de nuestro proyecto y los resultados que deseamos obtener y en que tiempo. Es por eso que propondremos una característica del proyecto y en base a ella evaluaremos cada proceso.

Tamaño de los equipos.

1. RUP: pensado para proyectos y equipos grandes, con roles designados y con una duración extendida.
2. XP: proyectos cortos con equipos pequeños y rotables en cuanto a roles.
3. FDD: pensado para proyectos y equipos medianos a pequeños, con la flexibilidad que a mayor necesidad de código, mayor organización.

Obtención de requisitos.

1. RUP: Se basa en los UseCase (casos de uso) donde se describen los requerimientos de la aplicación desde el punto de vista del usuario.

2. XP: Se Basa en los User Stories (historias de uso), que al igual que el anterior definen los detalles técnicos sin meterse con los detalles de implementación.
3. FDD: Define como se va a proceder después de recoger los requisitos definidos por el usuario.

Carga de trabajo.

1. RUP: proceso pensado por estar basado mucho en la documentación, la cual que se ve afectada con los posibles cambios volátiles que a los clientes se les ocurre en cuanto a funcionalidades del software, la justificación de esto es que gracias a su plan de desarrollo con el que se controla la evolución del sistema, se pueden reconocer los problemas y fallos de forma temprana y corregirlos.
2. XP: proceso ligero porque no se les asignan roles organizativos al equipo, roles como el modelado o generación de la documentación, esto es reemplazado por la presencia de un representante especializado del cliente, haciendo así más flexibles los posibles cambios que se presenten durante el desarrollo del software.
3. FDD: Aunque también genera documentación es la mínima necesaria para comprender el código, si bien es cierto el equipo cuenta con cierta libertad también es cierto que estos dependen directamente del jefe de proyecto quien es el responsable directo de proyecto.

Relación con el cliente.

1. RUP: Al final de cada fase, se les presenta al cliente los artefactos finales de dicha fase, para que sean evaluados por este y se puedan generar las iteraciones necesarias para la siguiente fase.
2. XP: La comunicación con el cliente es fluida (a través de su representante) después de cada iteración el cliente recibe un pedazo de programa funcional, así el cliente está informado permanentemente y puede intervenir rápidamente si el desarrollo se aleja de sus necesidades.
3. FDD: A diferencia del XP, aquí además de lo antes mencionado, existe el modelo general desarrollado en primera fase, la que provee un marco general dentro del cual evolucionará el proyecto (mientras no sea necesario cambiarlo claro)

Desarrollo.

Aquí los tres procesos están basados en iteraciones, lo que les permite acercarse poco a poco a la solución sin tener que entrar demasiado rápido a los detalles, la diferencia está en que los programadores de XP y FDD tienen menor carga a parte del desarrollo del software entonces les permite hacer las iteraciones con una menor duración.

Puntos flacos.

Es lógico pensar que si existen varios procesos de desarrollo de software es sencillamente porque cada uno presenta deficiencias o carencias según el punto de vista del equipo de desarrollo o las necesidades que el cliente presente, es por ellos que vamos a reconocer algunas desventajas de cada uno de estos tres procesos aquí mencionados.

1. RUP: Dado que este es un proceso general y cubre todas las posibles expectativas tanto del cliente como del equipo de desarrollo, es precisamente esa característica su mayor punto flaco, pues es muy grande para proyectos y equipos pequeños ya que deben repartirse 32 roles y generar muchos artefactos finales, los cuales pueden ser aprovechados en una reutilización de productos, modelos o procesos pero también significa un incremento de tiempos y costos.
2. XP: Este proceso está muy orientado a la implementación, esto lo hace bueno para el equipo de desarrollo ya que no debe preocuparse de la documentación y ya que los equipos rotan, todos aprenden de todos. Por otro lado el cliente también se siente satisfecho pues recibe un software que se adapta exactamente a sus deseos, para esto se debió designar a una persona totalmente involucrada en el negocio, lo que podría implicar que esta persona deje de hacer sus funciones para estar totalmente disponible al equipo de desarrollo, razón por la cual se considera mejor la utilización de este proceso para desarrollos internos, pues debe haber una gran confianza entre el cliente y el equipo de desarrollo. Como mencionamos antes era poco probable que el cliente pueda prescindir de sus empleados esto incurriría en un coste adicional para el cliente. Por último como podríamos representar todo lo que se debe sin documentación alguna (dependencia entre componentes por ejemplo) si no se anota ni se archiva nada, como alguien más puede tomar el lugar de uno de los miembros del equipo, o hacer mejoras en el sistema, esto crearía una dependencia para con el equipo de desarrollo.
3. FDD: El principal problema de este proceso esta representado por la necesidad de contar con miembros experimentados en el equipo, miembros que puedan definir los roles y funciones de cada uno, que marquen el camino a seguir desde el principio con la elaboración del modelo global, el hecho de ser una combinación de XP y RUP es lo que lo hace más atractivo a la implementación, claro siempre y cuando la experiencia de esa gente de mayor jerarquía no cree dependencias.

Programación de diseño

Java

Java es un lenguaje de programación con el que se puede realizar cualquier tipo de programa. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general. Está desarrollado por la compañía Sun Microsystems con gran dedicación y siempre enfocado a cubrir las necesidades tecnológicas más punteras. Es un lenguaje independiente de la plataforma; lo cual quiere decir que si se hace un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux, Apple, etc. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente.

La independencia de plataforma es una de las razones por las que Java es interesante para Internet, ya que muchas personas deben tener acceso con ordenadores distintos.

Actualmente Java se utiliza en un amplio abanico de posibilidades y casi cualquier cosa que se puede hacer en cualquier lenguaje se puede hacer también en Java y muchas

veces con grandes ventajas. Para lo que nos interesa a nosotros, con Java podemos programar páginas web dinámicas, con accesos a bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema. En general, cualquier aplicación que se desee hacer con acceso a través web se puede hacer utilizando Java.

Java fue diseñado para crear software altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Sus características de memoria liberan a los programadores de una familia entera de errores, ya que se ha prescindido por completo los punteros y la recolección de basura elimina la necesidad de liberación explicita de memoria.

C++

C++ es una mejoría sobre muchas de las características de C, y proporciona capacidades de POO que promete mucho para incrementar la productividad, calidad y reutilización del software.

Ventajas y desventajas de C++.

Ventajas:

- Al compilarlo, se genera código objeto, nativo de cada máquina. Resultado: C++ es más rápido que Java.

- Es una extensión de C. Por eso, muchos programadores encontrarán muy sencilla la transición, ya que podrán seguir haciendo cosas a la antigua usanza.

- Permite un control de la memoria y una capacidad de programación de bajo nivel impensable en Java.

Desventajas:

- No es multiplataforma. Para lograr aplicaciones que se ejecuten en varios SO, se requiere de cierto esfuerzo.

- No presenta una arquitectura estándar de desarrollo orientado a Internet. Java es algo más que un lenguaje, es toda una plataforma, y apoyada por muchas empresas, lo que le otorga un grado de calidad del que carece C++.

- Es una extensión de C. ¿Pero no era una ventaja? Bueno, pues también es un inconveniente, porque bastantes dogmas de la POO se sacrifican para hacer hueco al C. Java corrige esos problemas.

- No presenta un toolkit tan rico como el de Java. Este soporta el desarrollo rápido de aplicaciones, y muchas de las tareas de un programador están resueltas en su toolkit. Aunque hay muchas librerías en la red para C++, no son estándar del lenguaje, y algunas son de pago.

- Es más complicado de aprender que Java. También Java es complicado, y cuando se dice complicado se refiero a programar "bien" en Java, pero te obliga mucho más a seguir una metodología. C++, por ser en parte C, es demasiado libre en ocasiones.

Criterios de comparación para Java y C++.

- Expresividad: Facilidad del lenguaje para expresar los algoritmos

Java adopta una sintaxis muy similar a la del lenguaje C++, aunque eliminando algunas de sus características más oscuras. En particular, la eliminación de los punteros no lo ha hecho ni más ni menos expresivo, pero sí mucho más seguro.

- Bien definido: Consistencia y falta de ambigüedad

El lenguaje Java, fue creado desde el inicio con la intención de desterrar las ambigüedades y dependencias del implementador del lenguaje y de sus clases auxiliares, con lo cual actualmente es tal vez el mejor definido de los lenguajes populares.

- Tipos y estructuras de datos.

C++ proporciona facilidades que permiten la creación de estructuras de datos muy poderosas y fuertemente integradas en el lenguaje. Por ejemplo, las estructuras contenedoras "clásicas" se proporcionan en su librería de templates, la STL; asimismo, el desarrollador puede crear sus propios tipos de dato con diversas operaciones asociadas. Gracias a esto, su uso resulta una extensión natural de los tipos de dato primitivos con lo cual se alcanza un alto grado de claridad.

Java proporciona tipos de datos primitivos similares (notablemente, careciendo de punteros) y mediante su librería de clases estándar proporciona todas las estructuras contenedoras "clásicas" antes mencionadas, aunque con una sintaxis que pone claramente de manifiesto que se trata de clases auxiliares.

- Modularidad: permitir el desarrollo de componentes independientemente

Este criterio está referido a la posibilidad de desarrollar componentes de manera independiente los que eventualmente interactuarían. En ese sentido, los dos lenguajes analizados permiten desarrollar funciones, clases y paquetes de modo independiente, cada cual con sus convenciones particulares.

En cuanto a los "niveles de empaquetamiento" de los componentes, en C++ los conceptos de clase y "espacio de nombres" proporcionan dos niveles adicionales de "empacado", mientras que en Java los equivalentes corresponden a las clases y los "paquetes".

- Transportabilidad / Portabilidad

El lenguaje C++ siguió posteriormente un ciclo muy similar, y si bien no es un lenguaje automáticamente distribuido en los sistemas Unix, prácticamente todos lo pueden ejecutar ya sea en una variante comercial o mediante el popular GNU GCC/G++ con lo que la disponibilidad está asegurada. En cuanto a su portabilidad, el único inconveniente notorio radica en ciertos problemas (cada vez menos frecuentes) en las implementaciones de la STL.

No obstante lo indicado, el C++ presenta importantes dificultades de portabilidad, particularmente en cuanto a los siguientes aspectos:

1. Características dependientes de la implementación: Lo que permite realizar fuertes optimizaciones en distintas arquitecturas, resulta con frecuencia una pesadilla para la portabilidad. Muchos detalles importantes son dejados a criterio de quien escribe el compilador, tales como los tamaños de diversos tipos de datos, juegos de caracteres, comportamiento ante ciertos errores, etc.
2. Acceso a librerías del sistema operativo: Las interfaces y librerías principales no han seguido un proceso de estandarización tan riguroso como el lenguaje, lo que ha traído como consecuencia diversas soluciones incompatibles para los mismos problemas. Estrictamente este no es un problema del lenguaje, sino más bien de la plataforma utilizada (por ejemplo, las variantes de Unix.)

Estos problemas realmente nunca han tenido una solución definitiva, y a tal efecto existen algunas herramientas (por ejemplo, el "grupo" autoconf) orientadas mantenerlos "bajo control", mas no a eliminarlos. Asimismo, la escritura de un programa portable en C/C++ suele demandar la presencia de un programador experimentado que estructure adecuadamente el código a fin de facilitar el proceso de "portado" caso por caso.

En ese sentido Java introdujo un enfoque radical (aunque predecible) al diseñar un lenguaje prácticamente sin características dependientes del implementador (potencialmente algo menos eficiente), y con una extensa librería utilitaria cuya interfaz de programación está muy fuertemente estandarizada. Esto trajo consigo la famosa promesa: "write once, run everywhere" (escribir una sola vez, ejecutar en cualquier lugar) la cual ha sido muchas veces objeto de mofa debido a diversos errores de implementación y especificaciones poco claras ("write once, debug everywhere"). Con todo, la portabilidad alcanzada es cualitativamente superior a la que se puede obtener con los lenguaje C/C++, y se consigue de manera automática por cualquier desarrollador.

En conclusión, si es imprescindible una máxima portabilidad a "bajo costo", la respuesta es Java.

- ❖ Manipulación de archivos

C++ y Java proporcionan una interfaz alternativa a la misma funcionalidad a través de jerarquías de clases de I/O con diferente nivel de refinamiento, lo que los hace más extensibles aunque no necesariamente más convenientes. La tendencia en general apunta a no extender el lenguaje en este camino y por el contrario, crear nuevas librerías auxiliares para casos concretos.

- ❖ Acceso a Sistemas de Base de Datos

Un programa escrito en C++ tiene normalmente la capacidad de hacer uso del API de lenguaje C, pero muchos sistemas de base de datos proporcionan una interfaz mejorada orientada a objetos disponible en este lenguaje. Nuevamente, su uso no es portable.

Los creadores de Java, gracias a las anteriores experiencias, estandarizaron una interfaz orientada a objetos para acceder de un modo portable a cualquier base de datos. Esta API se denomina Java Database Connectivity (la conectividad de la base de datos Java) y

gracias a la gran popularidad de Java, prácticamente todos los vendedores importantes de bases de datos han creado implementaciones de esta interfaz. Esto promueve la portabilidad en cuanto al acceso a la base de datos, aunque las incompatibilidades y extensiones del SQL subsisten.

- ❖ No es importante ser veloz, sino no ser lento

Más allá de los benchmarks (puntos de referencias) y pruebas diversas de "cálculo puro" (en los que Java suele ser más lento que sus contendores) se suele plantear el argumento de la importancia de la velocidad de ejecución del lenguaje en sí. Si bien a todos les interesa que las aplicaciones se ejecuten a máxima velocidad, muchas veces la sensación de velocidad o lentitud no es ocasionada por la performance del "código principal" de la aplicación (que puede estar programado en Java) sino de componentes auxiliares tales como bases de datos, librerías de terceros, dispositivos gráficos acelerados, etc. En esa línea algunos defensores de Java manifiestan que es poco relevante si el "código C/C++" es 10 o 50% más veloz, si al final este tiempo no es el verdaderamente percibido por el usuario; asimismo, la aparente reducida performance de Java podría ser frecuentemente superada gracias a la claridad del lenguaje, el cual permitiría implementar mejores algoritmos y de un modo más eficiente.

Claramente, todos estos argumentos son subjetivos (pero muchas veces válidos) y al mismo tiempo son discutibles caso por caso. Un programa en Java suele ser notoriamente más lento si la tarea principal consiste en operaciones lógico/matemáticas, mientras que la performance suele ser ligeramente inferior a la correspondiente a C/C++ para aplicaciones que hacen uso de muchos otros componentes y librerías auxiliares.

- ❖ Pedagogía.

En breve, ni C ni C++ fueron creados para ser sencillos de aprender. C fue creado principalmente para ser eficiente, y C++ para ser a la vez eficiente y rico en características. Java, por el contrario tuvo desde el principio la intención de ser un lenguaje muy fácil de comprender y utilizar, y si bien eso no significa que su aprendizaje sea rápido ni trivial, ciertamente libera al estudiante de diversos aspectos confusos y sintaxis oscura de los otros lenguajes. Esta es quizá una de las razones más importantes que ha contribuido a su rápida adopción (aparte del excesivo marketing).

- ❖ Generalidad.

Los dos lenguajes estudiados se proponen como "de propósito general", es decir, serían adecuados para atacar prácticamente cualquier clase de problema. En la práctica, C++ y Java son utilizados para construir aplicaciones comerciales de toda clase. Notablemente Java, en gran medida gracias a la previsión y publicidad de Sun y diversos vendedores de "servidores de aplicación", es muy utilizado actualmente en el contexto de servidores Web (Servlets y JSP), acompañado en muchos casos de una arquitectura multicapa.

- ❖ Estandarización.

C++ es un buen ejemplo de lenguajes exitoso estandarizado "por comité" lo que promueve una competencia abierta entre las implementaciones, sin detrimento de la portabilidad.

Por su parte, Sun en sus inicios descartó utilizar un mecanismo similar para la estandarización de Java (lenguaje y librerías) pero luego dio paso a una apertura parcial en la que diversos vendedores y usuarios promueven los cambios en los estándares futuros, proceso que siempre es monitoreado por Sun (Java Comunity Program). Asimismo, Sun proporciona exigentes pruebas de certificación a fin de que los implementadores validen y publiciten su adherencia a los estándares, con el consiguiente beneficio de los desarrolladores.

- ❖ Evolución: ¿Qué esta ocurriendo con el lenguaje?

C++ continúa en su camino apuntando a una nueva revisión comúnmente conocida como C++0x (la idea es que aparezca antes de 2010) la cual estará más orientada hacia el desarrollo de las librerías (potencialmente -pero sin muchas probabilidades- incluyendo un API de GUI.) Este desarrollo es relativamente lento aparentemente debido a la falta de entusiasmo de los vendedores que suelen financiar esta clase de proceso.

Por su parte, Java continúa -a un paso acelerado- haciendo añadidos y mejoras en sus librerías principales y también en el lenguaje base (aunque estos últimos son contados) por lo general orientados a lograr una plataforma moderna y muy completa para distintas clases de aplicaciones.

- ❖ Soporte de librerías: ¿Qué NO se debe reescribir?

Por su parte, C++ dispone de una librería más extensa la cual incluye de hecho a la "librería estándar de C" así como la famosa "STL" (Standard Template Library) que implementa diversas estructuras de datos de manera genérica, así como muchos algoritmos populares. De igual modo, muchas librerías de terceros están disponibles para propósitos más especializados.

Java desde su creación tuvo la buena política de estandarizar muchas librerías (mediante clases e interfases) para una gran cantidad de aspectos que nunca fueron considerados en C ni C++ (como por ejemplo, la interfaz gráfica, el acceso a bases de datos, páginas Web, etc.) lo cual no excluye librerías más especializadas de terceros. Esto ha tenido un fuerte impacto en la comunidad de programadores que ven aquí una manera clara y segura de diseñar y codificar a partir de las especificaciones. El contexto de librerías estandarizadas en torno a Java es tan amplio que la "plataforma Java" se publicita como un conjunto de tecnologías orientas hacia distintos tipos de aplicaciones:

- Java SE (Java Standard Edition) considera facilidades de propósito general y aplicaciones de escritorio en particular
- Java EE (Java Enterprise Edition) para el desarrollo de aplicaciones empresariales (potencialmente sofisticadas) en servidores
- Java ME (Java Micro Edition) dirigida a la programación de dispositivos móviles.

De forma general:

Característica	**C++**	**Java**
Expresividad	Muy Buena a Excesiva	Muy Buena
Bien Definido	Muy Buena	Muy Buena

Característica	C++	Java
Tipos y Estructuras de datos	Muy Buena	Muy Buena
Modularidad	Muy Buena	Muy Buena
Facilidades de entrada/salida	Buena	Buena
Transportabilidad / Portabilidad	Buena	Excelente
Eficiencia / Performance	Excelente	Buena
Pedagogía	Regular	Buena
Generalidad	Muy Buena	Muy Buena
Estandarización	Buena	Excelente
Evolución	Estable	Acelerada
Soporte de Librerías	Muy Bueno	Excelente

Plataformas IDE para el desarrollo del sistema.

NetBeans

El IDE NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. Este IDE es un producto libre y gratuito sin restricciones de uso, soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles).

Características de Netbeans 6.0:

1.) Mejoras en el Code Editor:

a.) **Completamiento de Código**, más rápido e inteligente, mostrado de opciones y métodos mas rápido y mejorado.

b.) **Highlighting**: resaltar el código es una caracteristica fácil de usar y más correcta de sustituir el buscador de los editores. Los resaltes son marcados con un fondo de color distinto.

2.) Soporte de Lenguaje

a.) Soporte de JRuby/Ruby/Ruby on Rails – Excelente plantilla, auto-estructuración de carpetas. Muy buen depurador de Ruby. Una linda edición de archivos y highlighting RHTML.

b.) Mejorada la edición de Javascript y CSS. A pesar de que Dreamweaver es mejor, éste ha mejorado mucho a comparación de versiones anteriores.

c.) Soporte de lenguaje Schielmann. Es una tecnología que permite a cualquiera agregar soporte a nuevos lenguajes. Plugins para Php y Python se encuentran disponibles.

d.) Soporte de C/C++: Increíblemente mejorado el soporte para programar en C/C++. Las librerías son fácilmente accedidas y también tiene una muy buena edición de código.

3.) Interfaz Gráfica de Usuario (GUI) Swing:

a.) Aplicaciones de Base de Datos Swing. Probablemente la característica mas impresionante de NetBeans 6.0. Genial para hacer RAD para personas que quiere realizar rápidamente una aplicación de base de datos.

b.) Beans binding: Ahora puedes enmarcar beans y controlar su comportamiento desde la GUI. No necesitas el BDK (Beans Developement Kit) como un GUI para ver la edición del comportamiento de las beans y sus propiedades.

c.) Swing Application Framework (JSR 296): Espectacular soporte de aplicación de estado guardado. El estado de la aplicación es guardado por defecto. No necesitas preocuparte por el tamaño de la ventana de tu aplicación. Esta volverá con el mismo tamaño que tenía la última vez que la usaste, y para todo eso no se necesita escribir ni una sola línea de código.

4.) Profiling:

a.) Profiler Integrado- La herramienta de profiling con la que Netbeans es elegante y perfecto. Tal vez puedas compararla con LaodRunner, pero este hace más cosas en forma perfecta.

5.) Web & Java EE:

a.) Mejorado Visual Web (JSF): Después de que Sun decidiese que Studio Creator será netBeans, no hay mejor herramienta para hacer aplicaciones web JSF que netBeans 6.0. Simplemente arrastra y tira los componentes, escribe algo de código de lógica de negocios como EJBs y tendrás una aplicación web bien hecha.

b.) Soporte para AJAX habilitado componentes JSF.

c.) Muy buen editor de Javascript, con depurador de error, no visto en ningún otro editor WYSISWYG. Soporte CSS ha sido mejorado.

d.) Brillante administrador de flujo de aplicación Web. Ahora se puede realizar buenos diseños para páginas de flujo en netBeans 6.0.

6.) SOA:

a.) Editor Gráfico WSDL: NetBeans tiene una excelente herramienta visual para hacer todo el binding por nosotros mismos.

b.) Edición fácil de transformación en editor XSLT.

c.) Mejorada interoperabilidad con .Net a través de WSIT.

7.) UML:

a.) Generación de código mejorado: Menor diferencia en tus diseños y códigos.
b.) Brillantes mejoras en la edición de UML, a pesar de que algunas herramientas hace un mejor trabajo, es aún una gran característica para tener en su editor.

8.) Miscelánea:

a.) Mejorado dibujo en la plataforma del API.

b.) Un nuevo lexer parece hacer de NetBeans un aún más rápido editor de código.

Eclipse.

La plataforma Eclipse está diseñada para la construcción de entornos de desarrollo (IDEs) que puedan ser utilizados para la construcción de aplicaciones web, aplicaciones Java de todo tipo, programas C++ y Enterprise JavaBeans (EJBs). Este documento es una introducción a las posibilidades de la Plataforma Eclipse.

La Plataforma Eclipse está diseñada para afrontar las siguientes necesidades:

- Soportar la construcción de gran variedad de herramientas de desarrollo.
- Soportar las herramientas proporcionadas por diferentes fabricantes de software independientes (ISV's)
- Soportar herramientas que permitan manipular diferentes contenidos (i.e, HTML, Java, C, JSP, EJB, XML, y GIF).
- Facilitar una integración transparente entre todas las herramientas y tipos de contenidos sin tener en cuenta al proveedor.
- Proporcionar entornos de desarrollo gráfico (GUI) o no gráficos.
- Ejecutarse en una gran variedad de sistemas operativos, incluyendo Windows® y Linux™.
- Hacer hincapié en que el lenguaje de programación sea Java para la construcción de nuevos plugins.

El principal objetivo de la Plataforma Eclipse es proporcionar mecanismos, reglas que puedan ser seguidas por los fabricantes para integrar de manera transparente sus herramientas. Mediante APIs interfaces, clases y métodos, se exponen estos mecanismos. La Plataforma también posibilita la construcción de nuevas herramientas que extenderán sus funcionalidades.

Gestores de Base de Datos.

¿Qué es PostgreSQL?

- PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde la década de 1980.
- El proyecto PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto.
- PostgreSQL es ampliamente considerado como una de las alternativas de sistema de bases de datos de código abierto.

Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos

Características de PostgreSQL.

1. Soporte SQL92/SQL99: PostgreSQL implementa un subconjunto extendido de los estándares SQL92 y SQL99

2. Transacciones: Permiten el paso entre dos estados consistentes manteniendo la integridad de los datos.

3. Bloqueos de tabla y filas: Postgres ofrece varios modos de bloqueo para controlar el acceso concurrente a los datos en tablas Algunos de estos modos de bloqueo los adquiere PostgreSQL automáticamente antes de la ejecución de una declaración, mientras que otros son proporcionados para ser usados por las aplicaciones

4. Constraints y triggers: Tienen la función de mantener la integridad y consistencia en la BD. Ejecución de acciones antes o después de un evento de BD.

5. Múltiples tipos de datos predefinidos: Como todos los manejadores de bases de datos, PostgreSQL implementa los tipos de datos definidos para el estándar SQL3 y aumenta algunos otros.

6. Soporte de tipos y funciones de usuario: PostgreSQL soporta operadores, funciones métodos de acceso y tipos de datos definidos por el usuario.

7. Incorpora una estructura de datos Array: Conectividad TCP/IP, JDBC y ODBC. Interfaz con diversos lenguajes C, C++, Java, Delphi, Python, Perl, PHP, Bash.

8. Extensible: El código fuente está disponible para todos sin costo. Si su equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales. Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días.

9. Multiplataforma: PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y ahora en versión nativa para Windows.

Diseñado para ambientes de alto volumen PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC (Multi-Version Concurrency Control o Control de Concurrencia Multi-Versión) para conseguir una mejor respuesta en ambientes de grandes volúmenes. Los principales proveedores de sistemas de comerciales usan también esta tecnología, por las mismas razones.

Ventajas.

Las características positivas que posee este gestor según las opiniones más comunes en Internet, son:

1. Posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs 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 benchmarks se dice que ha llegado a soportar el triple de carga de lo que soporta MySQL).
2. Implementa el uso de rollback's, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz y ofreciendo soluciones en campos en las que MySQL no podría.
3. 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.

Desventajas.

Por contra, los mayores inconvenientes que se pueden encontrar a este gestor son:

1. Consume gran cantidad de recursos.
2. Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.
3. Es de 2 a 3 veces más lento que MySQL.

¿Qué es MySQL?

MySQL es un sistema de gestión de bases de datos relacional. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Características de MySQL.

Las principales características de este gestor de bases de datos son las siguientes.

1. Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
2. Soporta gran cantidad de tipos de datos para las columnas.
3. Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
4. Gran portabilidad entre sistemas.
5. Soporta hasta 32 índices por tabla.
6. Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

Ventajas.

Sin lugar a duda, lo mejor de MySQL es su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.

1. Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
2. Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
3. Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.
4. El conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet en servicios de foro (Barrapunto.com) y de buscadores de aplicaciones (Freshmeat.net).

Desventajas.

Debido a esta mayor aceptación en Internet, gran parte de los inconvenientes se exponen a continuación, han sido extraídos de comparativas con otras bases de datos:

1. Carece de soporte para transacciones, rollback's y subconsultas.
2. El hecho de que no maneje la integridad referencial, hace de este gestor una solución pobre para muchos campos de aplicación, sobre todo para aquellos programadores que provienen de otros gestores que sí que poseen esta característica.
3. No es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad.

Herramienta case para el modelado de la BD

Case Studio 2

Case Studio 2, permite crear y mantener Diagramas de Entidad Relación (DER) y diagramas de flujo de datos (DFD), crea scripts SQL de forma automática que funcionan sobre diferentes sistemas de bases de datos, además se puede utilizar plantillas para sistemas con características similares. Trata aspectos como contenidos y dominios para dar mejor seguridad al sistema. Contiene catálogos para guardar el modelo de diseño y soporte para ingeniería inversa que permite identificar y estructurar bases de datos ya existentes para poder trabajar con ellas. Facilita el trabajo de documentación al generar informes detallados en HTML y RTF, además de la documentación de Diagramas de Flujo de Datos exportables en formato XML. Soporta varios formatos de bases de datos como son, Oracle, MySQL, PostgreSQL, Access, MS SQL, Max DB, Firebird y muchos más. Tiene una interfaz grafica muy intuitiva y fácil de usar. Soportado en diferentes sistemas operativos, Windows 95/98/Me/NT/2000 /XP, además de ser una herramienta libre al igual que PostgreSQL.

EMS SQL Manager for PostgreSQL

EMS SQL Manager for PostgreSQL es una poderosa herramienta gráfica para la administración y desarrollo de PostgreSQL Database Server (servidor de bases de datos PostgreSQL). PostgreSQL Manager funciona con cualquier versión de PostgreSQL, hasta la 8.1, y soporta todas las nuevas características de PostgreSQL incluyendo espacios de tablas (tablespaces), argumentos nombrados (named arguments) en funciones y otras más. Ofrece una gran variedad de herramientas poderosas para usuarios avanzados, tales como Visual Database Designer (diseñador visual de base de datos), Visual Query Builder (constructor visual de consultas), y un poderoso editor de objetos binarios (BLOB) para satisfacer todas sus necesidades. PostgreSQL Manager cuenta con una nueva y avanzada interfaz gráfica de usuario con un sistema asistente bastante descriptivo, tan claro en su uso que ni un principiante se podrá confundir.

Características principales:

- Soporte completo de PostgreSQL hasta la versión 8.3
- Nueva interfaz gráfica de usuario último modelo
- Ágil navegación y administración de base de datos
- Administración sencilla de todos los objetos PostgreSQL
- Herramientas de manipulación de datos avanzada
- Administración efectiva de seguridad
- Excelentes herramientas visuales y de texto para elaboración de consultas
- Acceso al servidor PostgreSQL a través del protocolo HTTP
- Conexión vía reenvío de puerto local a través del túnel SSH
- Impresionantes opciones de exportación e importación de datos
- Poderoso diseñador visual de base de datos
- Asistentes fáciles de usar que realizan mantenimiento de tareas de PostgreSQL

Justificación de la propuesta de solución.

A partir de un análisis exhaustivo en este capítulo, como propuesta de solución se ha decidido por parte del equipo de desarrollo del sistema, desarrollar una aplicación de escritorio, siendo un *software* que resolvería las urgencias que presenta el Inder nacional, además permite mayor seguridad con la gestión de la información pues dadas las características de conecciones a la red en el país particularmente en el Inder.

Metodología.

El desarrollo de la propuesta de solución será guiada por la metodología eXtreme Programming (XP) que proponen un ciclo de desarrollo de software de manera ágil y novedosa, contando siempre con un representante del cliente interno que tendrá un contacto directo con el equipo de desarrollo, proporcionando así mayores posibilidades de desarrollar el sistema que realmente desea. Es la que mejor se adapta a nuestro caso de estudio y a las condiciones del equipo de desarrollo, teniendo en cuenta la propiedad colectiva del código o sea todos tienen autoridad para modificarlo según sus necesidades propias y del cliente en particular, haciéndose responsable de los cambios. La comunicación es abierta debido a que la programación es en parejas, esta es la mejor manera de entender como es el sistema y obtener un software de calidad sin muchas complicaciones o retrasos.

Lenguaje de programación.

Se propone que la implementación sea en el lenguaje Java dadas sus características y potencialidades, pues está totalmente basado en clases y objetos. Además al igual que otros lenguajes como C++ que son lenguajes multiplataforma, Java fue pensado desde un principio para ser adaptado a diferentes entornos, también maneja prácticamente todas las bases de datos relacionales, desde Microsoft SQL Server, Oracle, Sybase, Informix, como Access, MySQL y PostgreSQL.

Entorno de desarrollo

Se ha escogido el IDE NetBeans pues es un IDE una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso, soporta el desarrollo de todos los tipos de aplicación Java.

Gestor de Base de datos.

Proponemos el uso de PostgreSQL, servidor de base de datos relacional libre, liberado bajo una licencia de software libre.

La Herramienta manejadora de bases de datos a utilizar es: **PgAdmin III.**

PgAdmin III está diseñado para responder las necesidades de todos los usuarios, desde simples consultas SQL a la elaboración de bases de datos complejas. La interfáz gráfica de PostgreSQL soporta todas las características y hace fácil la administración. Es el más popular y rico en las características de código abierto de administración para PostgreSQL, la más avanzada base de datos de Open Source en el mundo.

Herramienta CASE para el modelado de la BD.

Como herramienta CASE para el modelado de la base de datos se usará **Case Studio 2,** que permite crear y mantener diagramas de entidad relación (DER) y diagramas de flujo

de datos (DFD), crea scripts SQL de forma automática que funcionan sobre diferentes sistemas de bases de datos, además se puede utilizar plantillas para sistemas con características similares. Trata aspectos como contenidos y dominios para dar mejor seguridad al sistema. Contiene catálogos para guardar el modelo de diseño y soporte para ingeniería inversa que permite identificar y estructurar bases de datos ya existentes para poder trabajar con ellas. Facilita el trabajo de documentación al generar informes detallados en HTML y RTF, además de la documentación de diagramas de flujo de datos exportables en formato XML. Soporta varios formatos de bases de datos como son, Oracle, MySQL, PostgreSQL, Access, MS SQL, Max DB, Firebird, y muchos más. Tiene una interfaz grafica muy intuitiva y fácil de usar. Soportado en diferentes sistemas operativos, Windows 95/98/Me/NT/2000 /XP, además de ser una herramienta libre al igual que PostgreSQL.

Arquitectura.

La arquitectura se basa entonces en 3 capas principales:

- Capa de presentación.
- Capa de lógica de negocio.
- Capa de acceso a datos.

Capa de presentación.

Esta capa resuelve la presentación de datos al usuario, se encarga entonces de "dibujar" las pantallas de la aplicación al usuario, y tomar los eventos que el cliente genere (por ejemplo, el hacer click en un botón).
Existen numerosas tecnologías para esta capa (JSP, Struts, JSF, Flex, Velocity, Wicket, GWT, etc, etc), cada una con sus ventajas y desventajas.

Capa de lógica del negocio.

Esta capa resuelve la lógica de la aplicación. Contiene los algoritmos, validaciones y coordinación necesaria para resolver la problemática.
Los elementos fundamentales de esta capa son los **objetos de dominio**, los cuales representan los objetos principales del negocio. Son simples objetos que sólo contienen los datos que representan (por ejemplo, un Cliente, una Factura, una Direccion, un Producto).
La lógica para manipularlos se encuentra en los llamados objetos de negocio (**Business Object**, o BO). Estos objetos contienen la lógica del negocio, y manipulan los objetos del dominio. A su vez, estos son los objetos que exponen métodos que se transforman en el contrato de la capa de lógica de negocio.

Los BO constan de una interfáz (dónde se expone la lógica del negocio) y una clase que la implementa.

Capa de acceso a datos.

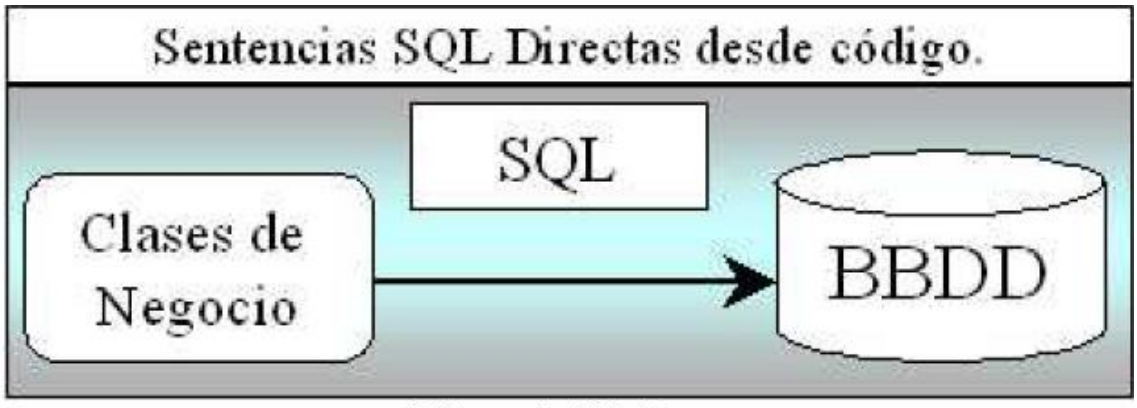
Esta capa resuelve el acceso a datos, abstrayendo a su capa superior de la complejidad del acceso e interacción con los diferentes orígenes de datos. Se encarga de proveer un API simple de usar, orientado al negocio, sin exponer complejidades propias de un repositorio de datos.

En ella se resuelven:

- Cualquier acceso a la base de datos.
- Cualquier acceso a file system.
- Cualquier acceso a otros sistemas.
- Cualquier acceso a un repositorio de datos en cualquier forma.

Estos objetos se construyen a partir del patrón Data Access Object (**DAO**). Los DAO constan de una interfáz (dónde se expone la lógica de acceso a datos) y una clase que la implementa.

En el diseño de una aplicación a desarrollar utilizando Java una parte muy importante es la manera en la cual accedemos a nuestros datos en la base de datos (en adelante BD) determinar esta parte se convierte en un punto crítico para el futuro desarrollo. La manera tradicional de acceder seria a través de JDBC directamente conectado a la BD mediante ejecuciones de sentencias SQL:

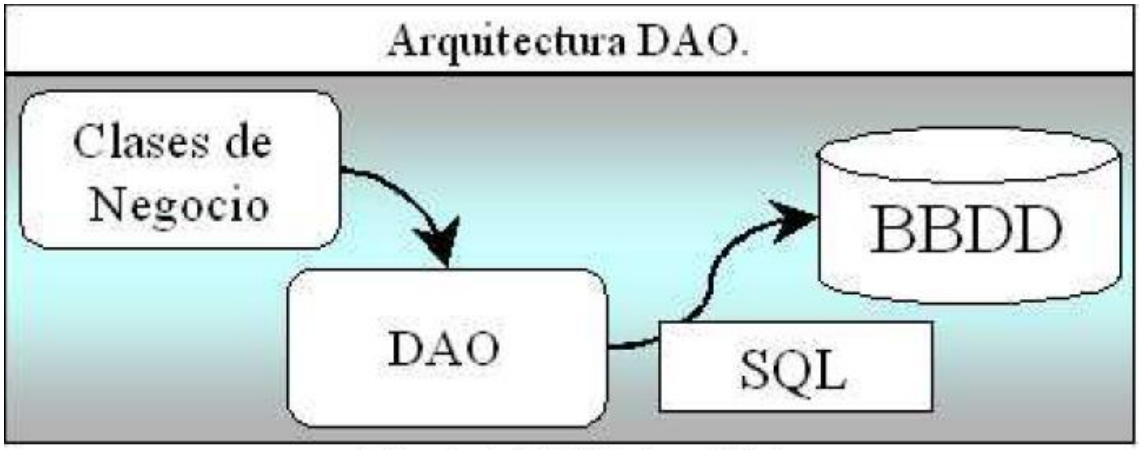


1. Sentencias SQL directas

Esta primera aproximación puede ser útil para proyectos o arquitecturas sin casi clases de negocio, ya que el mantenimiento del código está altamente ligado a los cambios en el modelo de datos relacional de la BD, un mínimo cambio implica la revisión de casi todo el código así como su compilación y nueva instalación en el cliente. El acceso a través de SQL directas puede ser utilizado de manera puntual para realizar operaciones a través del

lenguaje SQL lo cual sería mucho más efectivo que la carga de gran cantidad de objetos en memoria. Si bien un buen motor de persistencia debería implementar mecanismos para ejecutar estas operaciones masivas sin necesidad de acceder a este nivel.

Una aproximación más avanzada seria la creación de unas clases de acceso a datos (**DAO**). De esta manera nuestra capa de negocio interactuaría con la capa DAO y esta sería la encargada de realizar las operaciones sobre la BD.



2. Ejemplo de DAO (Data Access Object

Los problemas de esta implementación siguen siendo el mantenimiento de la misma así como su portabilidad. Lo único que se puede decir es que se tiene el código de transacciones encapsulado en las clases DAO. Un ejemplo de esta arquitectura podría ser Microsoft ActiveX Data Object (ADO).

Conclusiones.

En este capítulo se realizó un estudio exhaustivo de las tendencias actuales de las distintas tecnologías que se utilizarán para la realización del caso de estudio en cuestión, así como una serie de conceptos que se deben dominar para el entendimiento del mismo. Y para culminar el estudio se realizó la selección de las tecnologías, lenguajes, y metodología a usar para la solución del caso de estudio en cuestión.

Capítulo 2: Exploración y Planificación.

Introducción

En este capítulo se hace alusión a las fases de exploración y planificación correspondientes a la metodología de desarrollo utilizada para la implementación de un sistema que cubra todas las necesidades del cliente. Las características de dicho sistema serán definidas luego de obtener las historias de usuario (HU), creadas por el representante del cliente y estudiadas por el equipo de desarrollo, de forma tal que cumpla con las expectativas del equipo y las necesidades del cliente. Se exponen además los artefactos que se generan en el transcurso de la misma.

Fase de exploración.

La metodología de desarrollo eXtreme Programming comienza con la fase de exploración, durante la cual se realiza el proceso de identificación de las HU, así como la propuesta y definición de un sistema que cumpla con todos los requisitos necesarios para hacer cumplir cada una de ellas, así como la familiarización del equipo de trabajo con las tecnologías y herramientas seleccionadas para la construcción del sistema.

Historias de Usuario (HU).

Las HU son la forma en que se especifican en XP los requisitos del sistema. Estas son redactadas por el cliente o por su representante aunque los desarrolladores pueden brindar su ayuda en la identificación de las mismas. El contenido que ellas abarcan debe ser concreto y sencillo. Durante el presente proceso se identifican 10 HU las cuales se detallan posteriormente.

Estas Historias de Usuario a su vez producen tareas de ingeniería, las cuales tienen que ver más directamente con las acciones que puede o no desarrollar el programador para dar respuesta a la misma. Una vez cumplida una tarea, se realizan un conjunto de pruebas de aceptación para asegurarse que los componentes desarrollados funcionan debidamente y una prueba de integración para probar el desarrollo en su interacción con los demás componentes implementados. Estas tareas serán documentadas en capítulos posteriores.

Para la creación de las historias de usuario se utilizó una plantilla la cual es un estándar que recoge toda la información necesaria y bastante bien explicado para que el cliente o su representante la puedan llenar sin ningún problema. También se tiene en cuenta la opinión del programador dándole la posibilidad de determinar según las prioridades del cliente en que iteración se realizará y la complejidad que tiene la misma, así como el tiempo que se demorará en realizarla.

Relación de las Historias de Usuarios.

1. HU Asegurar Evento.

Historia de Usuario	
Número: 1	**Usuario: Comisión aseguradora.**
Nombre historia: Asegurar Evento.	
Prioridad en negocio: Alta	**Riesgo en desarrollo: Alto**
Puntos estimados: 1	**Iteración asignada: 1**
Programador responsable: Darien Concepción y Martha L. Gala	
Descripción: Se le brinda la posibilidad a La Comisión Aseguradora de introducir los datos característicos del evento, así como los necesarios para realizar las operaciones deseadas desarrolladas en HU posteriores, guardando todos los datos en el sistema.	
Observaciones:	

2. HU Gestionar Evento

Historia de Usuario	
Número: 2	**Usuario: Comisión aseguradora.**
Nombre historia: Gestionar Evento.	
Prioridad en negocio: media	**Riesgo en desarrollo: Alto**
Puntos estimados: 1	**Iteración asignada: 1**
Programador responsable: Darien Concepción y Martha L. Gala	
Descripción: Se le brinda la posibilidad a La Comisión Aseguradora de modificar, eliminar y mostrar los datos característicos del evento.	

Observaciones:

3. HU Gestionar Inscripción de atletas.

Historia de Usuario	
Número: 3	**Usuario: Comisión organizadora.**
Nombre historia: Gestionar Inscripción de atletas.	
Prioridad en negocio: Alta.	**Riesgo en desarrollo: Alto.**
Puntos estimados: 2	**Iteración asignada: 2**
Programador responsable: Darien Concepción y Martha L. Gala	
Descripción: Se brinda la posibilidad al encargado de La Comisión Organizadora de insertar, modificar, eliminar y mostrar todos los datos del atleta. Datos como nombre completo, modalidad, No CI, Grupo sanguíneo, nacionalidad, marcas personales en la temporada, record olímpico, record mundial.	
Observaciones:	

4. HU Archivar Datos de la competencia.

Historia de Usuario	
Número: 4	**Usuario: Estadístico.**
Nombre historia: Archivar Datos de la competencia.	
Prioridad en negocio: Alta.	**Riesgo en desarrollo: Alto**
Puntos estimados: 3	**Iteración asignada: 3**
Programador responsable: Darien Concepción y Martha L. Gala	
Descripción: Se le brinda la posibilidad al estadístico de poder insertar los datos generados durante	

la competencia en cualquier modalidad.
Observaciones: Cada modalidad tiene una interfáz correspondiente con las unidades de medida correspondientes a cada evento.

5. HU Gestionar Datos de la competencia.

Historia de Usuario	
Número: 5	**Usuario: Estadístico.**
Nombre historia: Gestionar Datos de la competencia.	
Prioridad en negocio: Alta.	**Riesgo en desarrollo: Alto**
Puntos estimados: 3	**Iteración asignada: 3**
Programador responsable: Darien Concepción y Martha L. Gala	
Descripción: Se le brinda la posibilidad al estadístico de poder modificar, eliminar y mostrar, los datos generados durante la competencia en cualquier modalidad.	
Observaciones:	

6. HU Modalidad y Títulos.

Historia de Usuario	
Número: 6	**Usuario: Comisión organizadora.**
Nombre historia: Modalidad y Títulos	
Prioridad en negocio: Media	**Riesgo en desarrollo: Medio**
Puntos estimados: 2	**Iteración asignada: 3**
Programador responsable: Darien Concepción y Martha L. Gala	

Descripción: Se brinda la posibilidad al encargado de La Comisión Organizadora de visualizar todos los datos correspondientes a la modalidad y título. Datos como, disciplina (campo y pista, maratón, exhibición), modalidad, cantidad de pruebas por sexo, títulos individuales y por equipos, cantidad de medallas físicas (oro, plata, bronce).
Observaciones: Estos datos son generados desde la BD y se muestran en una tabla con los valores correspondientes.

7. HU Ranking por modalidad.

Historia de Usuario	
Número: 7	**Usuario: Estadístico**
Nombre historia: Ranking por modalidad.	
Prioridad en negocio: Baja	**Riesgo en desarrollo: Medio**
Puntos estimados: 1	**Iteración asignada: 4**
Programador responsable: Darien Concepción y Martha L. Gala	
Descripción: Se le brinda la posibilidad al estadístico de visualizar un ranking por modalidad para los atletas permitiéndole ver los listados de dichos ranking.	
Observaciones: Estos datos son generados de la BD y mostrados en una tabla.	

8. HU Ranking por países.

Historia de Usuario	
Número: 8	**Usuario: Estadístico**
Nombre historia: Ranking por países.	

Prioridad en negocio: Baja	**Riesgo en desarrollo: Medio**
Puntos estimados: 1	**Iteración asignada: 4**
Programador responsable: Darien Concepción y Martha L. Gala	
Descripción: Se le brinda la posibilidad al estadístico de visualizar un ranking por puntos para los países permitiéndole ver los listados de dichos ranking.	
Observaciones: Estos datos son generados de la BD y mostrados en una tabla.	

9. HU Autenticar Usuario.

Historia de Usuario	
Número: 9	**Usuario: Usuario.**
Nombre historia: Autenticar Usuario.	
Prioridad en negocio: Media	**Riesgo en desarrollo: Bajo**
Puntos estimados: 1	**Iteración asignada: 4**
Programador responsable: Darien Concepción y Martha L. Gala	
Descripción: Se le brinda la posibilidad al usuario de poder autenticarse en el sistema.	
Observaciones:	

10. HU Gestionar usuario.

Historia de Usuario	
Número: 10	**Usuario: Administrador**
Nombre historia: Gestionar usuario.	
Prioridad en negocio: Baja	**Riesgo en desarrollo: Bajo**

Puntos estimados: 1	**Iteración asignada: 4**
Programador responsable: Darien Concepción y Martha L. Gala	
Descripción: Se le brinda la posibilidad al administrador de poder insertar, modificar y eliminar un usuario en el sistema.	
Observaciones:	

Después de un exhaustivo estudio de las HU que plantea el representante del cliente se llega a la siguiente conclusión.

Objeto de automatización.

El Sistema de Gestión de Informatización del Deporte SGID, tiene como objetivo principal la automatización de un submódulo del sistema completo, el cual estará relacionado con el deporte de Atletismo en todas sus modalidades. Esto sería la recopilación de los datos personales correspondiente a todos los participantes del evento deportivo que se esté desarrollando, así como los tiempos y las distancias realizadas por cada atleta en su modalidad correspondiente, además de brindar la posibilidad de consultar el reglamento o emitir una nueva reclamación.

Sistema propuesto.

El presente Trabajo de Diploma propone la implementación de un sistema que provea todas las funcionalidades que se deben automatizar descritas anteriormente. Dicho sistema debe permitir la gestión (insertar, eliminar, modificar) de los datos correspondientes a los participantes del evento, así como mostrar toda la información debida respecto todos los datos generados y archivados durante el evento propiamente dicho.

Con la implementación del sistema propuesto los eventos deportivos de atletismo que se celebren ganarán calidad y eficiencia, ya que se facilita el trabajo de los responsables de que este encuentro deportivo para que se realice lo mejor posible, pues con dicho sistema se podrá eliminar todo el proceso manual, además de lograr una mejor organización e información a todos los espectadores que deseen ver los resultados de las competencias con solo acceder a una aplicación Web que será conectada a la BD (no está dentro de nuestro campo de acción) para obtener toda la información que desee conocer sobre el evento. Podrá ver los listados con todos los datos personales del atleta, así como los tiempos y/o distancias obtenidos en el transcurso de la competencia. Como estos datos son recogidos instantáneamente durante el transcurso del evento, el usuario podrá verlos inmediatamente que sean generados en el calor del evento propiamente dicho.

El estadístico es otro trabajador que tendrá mucho beneficio con la implementación de este sistema pues también dejará a un lado el tedioso trabajo de buscar entre todos los papeles el nombre de un atleta determinado para registrar el tiempo realizado en una carrera, lanzamiento o salto. Con el sistema propuesto sólo tendrá que buscar el nombre del atleta en un formulario, que estará conformado según los participantes en ese momento y en esa modalidad, e introducir el dato generado instantáneamente. También se le brinda la posibilidad de generar el ranking por modalidades según el sexo, y por países según la puntuación.

El sistema será una aplicación de escritorio implementada mediante el lenguaje Java, creando así un sistema sencillo y seguro, ya que si existen problemas de conexiones en la red, los datos generados se podrán seguir guardando en la base de datos soportada por el gestor de base de datos PostgreSQL y cuando las conexiones regresen a la normalidad el usuario podrá acceder fácilmente a los datos generados durante el transcurso de la competencia.

Apariencia del sistema.

Como característica propia de la metodología XP el diseño del sistema deberá ser fácil de implementar y sencillo de entender, permitiendo una mejor familiarización entre el mismo y los usuarios, además de no necesitar una compleja capacitación para el uso del mismo. No debe llevar sobrecarga de campos a llenar, para que al usuario del sistema no se le haga engorroso el trabajo que debe realizar. No debe tener opciones adicionales sin que se utilicen para no sobrecargar el diseño o apariencia del sistema (Ver anexo 1 y 2)

Fase de planificación.

Durante la presente fase se realiza una estimación del esfuerzo que costará implementar cada HU. Esto se expresa utilizando como medida el punto, este se considera como una semana de ideal de trabajo, donde cada miembro del equipo de desarrollo trabaja el tiempo planeado sin ninguna interrupción. Esta estimación incluye todo el esfuerzo asociado a la implementación de una HU, por ejemplo, las pruebas unitarias, la integración, la refactorización del código y la preparación y ejecución de las pruebas de aceptación.

Estimación de esfuerzos.

Para lograr un desarrollo eficiente y satisfactorio, se realizó una estimación de esfuerzos para cada una de las HU identificadas en el proceso de planificación, llegando a los resultados que se muestran a continuación.

Tabla 13: Estimación de esfuerzos.

Historias de Usuario	Puntos de Estimación.
Asegurar Evento.	1
Gestionar Evento.	1
Gestionar inscripciones de Atletas.	1
Archivar Datos de la competencia.	3
Gestionar Datos de la competencia.	1

Modalidad y Títulos.	1
Ranking por modalidad.	1
Ranking por países.	1
Autenticar Usuario.	1
Gestionar Usuario.	1

Plan de iteraciones.

Una vez descritas e identificadas las HU y estimado el esfuerzo propuesto para la realización de cada una de ellas, se procede a la planificación de la etapa de implementación del sistema. Este plan especifica exactamente cuales son las HU que serán implementadas para cada iteración y las posibles fechas de entregas. Esta decisión se tomó a partir de la importancia en el negocio y por el orden lógico de acceso a cada acción que deberá realizar para darle cumplimiento a cada una de las HU. No se tuvo en cuenta ningún otro criterio ya que se consideró que estos eran suficientes para determinar la factibilidad y la importancia de una HU.

En base a lo antes dicho se decide realizar cuatro iteraciones las cuales se dividen de la siguiente forma:

Iteracion 1

Esta iteración tiene como objetivo la implementación de las HU 1, 2, 3, con prioridades alta, media y alta respectivamente, aunque la prioridad puede ser media no quiere decir que no sea altamente importante para el cliente ya que hace alusión a la organización completa del evento o sea permiten identificar el evento según su descripción además de insertar correctamente los datos personales de los atletas. Luego de implementar las HU se tendrá la primera versión del sistema propuesto, la cual será mostrada al cliente, con el objetivo de la retroalimentación con el equipo de desarrollo además de tener la posibilidad de poder pasar a la siguiente iteración.

Iteración 2

Aquí se tiene como idea principal la implementación de la HU 4 la cual es conjuntamente con la HU No 3 las más importante y compleja, ya que corresponden al problema principal al que se le está dando solución. Después de ser desarrollada se tendrá la segunda versión del sistema propuesto, y conjuntamente con la versión anterior, será mostrada al cliente con el objetivo de aprobación o cambios pertinentes de parte del cliente. Además se podrá tomar la decisión de pasar a la próxima iteración.

Iteración 3

Aquí se raliza la implementación de las HU 5, 6, 7 las cuales tienen como objetivo principal gestionar todos los datos que son generados por los atletas durante la competencia, generar las tablas relacionadas con las modalidades desarrolladas y títulos obtenidos, así como el ranking por modalidad para los atletas participantes. Una vez finalizada la implementación se obtendrá una porción casi media del sistema que se quiere implementar para satisfacción del cliente, quien aprobará o no los resultados obtenidos de la presente iteración. En caso de aprobarla tomará la decisión de comenzar con la próxima iteración.

Iteración 4

En esta iteración se completará el sistema propuesto con la implementación de las HU 8, 9, 10 correspondientes a generar la tabla del ranking por paises, y las funcionalidades que tienen que ver con la seguridad del sistema. Estas HU permitirán tener un 100 % de la aplicación implementada y disponible para ser probada por el cliente en su totalidad.

Plan de duración de las iteraciones.

Como parte del ciclo de vida de un proyecto que está utilizando la metodología XP se crea el plan de duración de iteraciones, en este caso se hace para el único equipo de desarrollo con el que se cuenta para la implementación de este submódulo de Atletismo, del módulo completo para La Gestión de Informatización del Deporte. Este plan tiene como finalidad mostrar la duración de cada iteración, así como el orden en que serán implementadas las historias de usuario de cada una de las mismas.

Tabla 14: Plan de duración de iteraciones.

No Iteración	Historias de Usuario	Duración total de la Iteración
Iteración 1	- Asegurar Evento. - Gestionar Evento. - Gestionar Inscripción de atletas.	3 Semanas.
Iteración 2	- Archivar Datos de la Competencia.	3 Semanas.
Iteración 3	- Gestionar Datos de la Competencia. - Gestionar Modalidad y títulos. - Ranking por modalidad.	3 Semanas.
Iteración 4	- Ranking por países. - Autenticar Usuario. - Gestionar Usuario	3 Semanas.

Plan de entregas.

A continuación se presenta el plan de entregas desarrollado para dar solución al problema planteado, la misma está compuesto por la cantidad de iteraciones y la fecha de entrega, para la cual se tuvo en cuenta los puntos de estimación sumando los mismos y obteniendo un resultado final.

Tabla 15: Plan de entregas del sistema propuesto.

No. de iteración	Duración total de la iteración.	Fecha de inicio.	Fecha de fin.
Iteración 1	3 semanas	11/02/09	29/02/09
Iteración 2	3 semanas	01/03/09	25/03/09
Iteración 3	3 semanas	01/04/09	25/04/09
Iteración 4	3 semanas	30/04/09	18/05/09

Conclusiones.

Durante todo este capítulo se abordó toda lo información correspondiente a la fase de exploración y planificación conformando así la mayor parte de los artefactos correspondientes a esta fase y tratando de organizar lo mejor posible el trabajo tanto para el equipo de desarrollo como para el entendimiento del cliente o su representante, con el único objetivo de lograr el sistema deseado para poder satisfacer cabalmente la necesidad del cliente.

Capítulo 3: Diseño y Prueba.

Introducción.

La metodología X.P sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implementable que a la larga costará menos tiempo y esfuerzo. Es por eso que en este capítulo se tratará de encontrar la mejor manera de obtener un diseño simple; para obtener un resultado totalmente óptimo, trabajamos con las tarjetas Clase-Responsabilidades-Colaboradores, conocidas como tarjetas CRC. Además se realizaron conjuntamente con el cliente las pruebas funcionales que sirven de paso como pruebas de aceptación, las cuales dan una seguridad de que el código implementado del sistema será el que verdaderamente hace falta, y que además deberá pasar para poder tomarse como código valido. Se desarrollaron además tareas de ingeniería en las HU para un mejor entendimiento y apoyo para el equipo de desarrollo.

Tareas de Ingeniería.

Estas tareas de ingeniería se realizan para desglosar mejor la HU en caso de que sea necesario para su mejor entendimiento e implementación. Se le puede hacer todas las tareas que se deseen a cada HU que lo necesite o sea no todas llevan necesariamente tareas de ingeniería. En cada tarea se explicará detalladamente lo que hace en ese paso la HU. Los puntos estimados en cada tarea fueron asignados en días según la complejidad del mismo y sumando dichos puntos de cada tarea de ingeniería se obtendrá la semana completa que se demora la implementación de la HU en su totalidad. Para la realización de estas tareas se utilizó una plantilla estándar implementada en La Universidad la cual se muestra a continuación:

Tarea	
Número tarea:	**Número historia:**
Nombre tarea:	
Tipo de tarea : Desarrollo \ Corrección \ Mejora \ (otra Especificar)	**Puntos estimados:**
Fecha inicio:	**Fecha fin**:
Programador responsable:	
Descripción:	

Tabla 16: Plantilla de Tareas de Ingeniería.

A continuación se describen algunas de las tareas de ingenieria realizadas para algunas HU:

Tarea	
Número tarea: 1	**Número y nombre historia**: HU 1: Asegurar Evento
Nombre tarea: Entrar Datos.	
Tipo de tarea : Desarrollo	**Puntos estimados: 2**
Fecha inicio: 11/02/09	**Fecha fin**:13/02/09
Programador responsable: Darién Concepción y Martha L. Gala	
Descripción: Una vez insertados los datos correspondientes al evento a celebrar y pulsado el botón aceptar, se verificarán que los datos insertados sean válidos, o sea que no se inserten valores incorrectos en los distintos campos de texto.	

✂----------

Tarea	
Número tarea: 2	**Número y nombre historia:** HU 1: Asegurar Evento.
Nombre tarea: Verificar datos.	
Tipo de tarea : Desarrollo	**Puntos estimados:2**
Fecha inicio: 14/02/09	**Fecha fin**: 16/02/09
Programador responsable: Darién Concepción y Martha L. Gala	
Descripción: Una vez insertados y validados los datos se verifica en la Base de Datos si existen o no, con el objetivo de no tener repetidos los datos del evento. En caso de que los datos sean válidos o no, se enviará un mensaje de notificación o verificación correspondientemente.	

✂----------

Tarea	
Número tarea: 1	**Número y nombre historia:** HU 2: Gestionar Evento.

Nombre tarea: Modificar datos del evento.	
Tipo de tarea : Desarrollo	**Puntos estimados: 2**
Fecha inicio: 17/02/09	**Fecha fin**: 20/02/09
Programador responsable: Darién Concepción y Martha L. Gala	
Descripción: Se muestra un listado de todos los eventos y al seleccionar el que se desea modificar se muestra un listado con sus datos individuales y los campos disponibles para realizar los cambios pertinentes.	

✂----------

Tarea	
Número tarea: 2	**Número y nombre historia:** HU 2: Gestionar Evento.
Nombre tarea: Eliminar datos del evento.	
Tipo de tarea : Desarrollo	**Puntos estimados:** 1
Fecha inicio: 21/02/09	**Fecha fin**: 22/02/09
Programador responsable: Darién Concepción y Martha L. Gala	
Descripción: Se muestra un listado de eventos y al seleccionar el evento que se desea eliminar se mostrará un listado con todos sus datos individuales y la opción eliminar. Al pulsar esta opción se eliminarán de la BD todos los datos de este evento y se envía un mensaje de ratificación.	

✂----------

Tarea	
Número tarea: 1	**Número y nombre historia:** HU 3: Gestionar inscripción de atletas
Nombre tarea: Insertar datos del atleta.	
Tipo de tarea : Desarrollo	**Puntos estimados: 2**
Fecha inicio: 01/04/09	**Fecha fin**: 02/04/09
Programador responsable: Darién Concepción y Martha L. Gala	
Descripción:	

Se muestra una interfáz que brinda los campos necesarios para registrar los datos del atleta. Al pulsar el botón de aceptar se validan si los datos y en caso de errores se ratifica al usuario.

✂----------

Tarea	
Número tarea: 2	**Número y nombre historia:** HU 3: Gestionar inscripción de atletas.
Nombre tarea: Validar entrada de datos.	
Tipo de tarea : Desarrollo	**Puntos estimados: 1**
Fecha inicio: 03/04/09	**Fecha fin:** 03/04/09
Programador responsable: Darién Concepción y Martha L. Gala	
Descripción: Todos los datos que son entrados son verificados y validados, en caso de que los datos sean entrados con errores el usuario será ratificado mediante un mensaje y se le brindará la posibilidad de insertar una vez más los datos correctamente. En caso de que los datos que fueron insertados ya existen en la BD implementada, se mostrará un mensaje de información al cliente.	

✂----------

Tarea	
Número tarea: 3	**Número y nombre historia:** HU 3: Gestionar inscripción de atletas
Nombre tarea: Modificar datos del atleta.	
Tipo de tarea : Desarrollo	**Puntos estimados:** 2
Fecha inicio: 04/04/09	**Fecha fin**: 06/04/09
Programador responsable: Darién Concepción y Martha L. Gala	
Descripción: Para modificar los datos del atleta se mostrará inicialmente un listado con los nombres de todos los que fueron inscritos y al seleccionar uno de ellos se mostrará más en detalles sus datos personales y los campos correspondientes disponibles para realizar las modificaciones.	

✂----------

Tarea

Número tarea: 4	**Número y nombre historia:** HU 3: Gestionar inscripción de atletas.
Nombre tarea: Eliminar datos del atleta.	
Tipo de tarea : Desarrollo	**Puntos estimados:** 1
Fecha inicio: 07/04/09	**Fecha fin**: 08/04/09
Programador responsable: Darién Concepción y Martha L. Gala	
Descripción: Para eliminar los datos del atleta se mostrará inicialmente un listado con los nombres de todos los que fueron inscritos y al seleccionar uno de ellos se visualiza más en detalles sus datos personales y la opción de eliminar y al seleccionarla se eliminarán totalmente de la BD todo lo correspondiente a este participante.	

✂----------

Tarea	
Número tarea: 1	**Número y nombre historia:** HU 4: Archivar datos generados.
Nombre tarea: Insertar datos.	
Tipo de tarea : Desarrollo	**Puntos estimados:** 2
Fecha inicio: 18/04/09	**Fecha fin**: 19/04/09
Programador responsable: Darién Concepción y Martha L. Gala	
Descripción: Se selecciona la modalidad a la cual se le van a archivar los datos generados por los atletas, luego se muestra un listado con los atletas participantes en la misma y los campos necesarios disponibles para la inserción de los datos.	

✂----------

Tarea	
Número tarea: 2	**Número y nombre historia:** HU 4: Archivar datos generados
Nombre tarea: Validar datos de la especialidades de distancia.	
Tipo de tarea : Desarrollo	**Puntos estimados:** 2
Fecha inicio: 20/04/09	**Fecha fin**: 22/04/09

Programador responsable: Darién Concepción y Martha L. Gala
Descripción: Todos los datos entrados serán validados ya que solo se permitirá el siguiente formato X.X

✂----------

Tarea	
Número tarea: 3	**Número y nombre historia:** HU 4: Archivar datos generados
Nombre tarea: Validar datos de la especialidades de tiempo.	
Tipo de tarea : Desarrollo	**Puntos estimados:** 2
Fecha inicio: 23/04/09	**Fecha fin**: 24/04/09
Programador responsable: Darién Concepción y Martha L. Gala	
Descripción: Todos los datos entrados serán validados ya que solo se permitirá el siguiente formato Xx:Xx:Xx.	

✂----------

Tarea	
Número tarea: 1	**Número y nombre historia:** HU 5: Gestionar datos de la competencia.
Nombre tarea: Modificar datos.	
Tipo de tarea : Desarrollo	**Puntos estimados: 2**
Fecha inicio: 30/04/09	**Fecha fin**: 01/05/09
Programador responsable: Darién Concepción y Martha L. Gala	
Descripción: Se selecciona la modalidad y se muestra un listado con los nombres de los atletas participantes en ese evento y los campos correspondientes con el tiempo o la distancia (en dependencia de la modalidad), disponibles para realizar los cambios necesarios.	

✂----------

Tarea	
Número tarea: 2	**Número y nombre historia:** HU 5: Gestionar datos de la competencia.

Nombre tarea: insertar datos.	
Tipo de tarea : Desarrollo	**Puntos estimados:** 1
Fecha inicio: 02/05/09	**Fecha fin**: 03/05/09
Programador responsable: Darién Concepción y Martha L. Gala	
Descripción: Se verifica que los datos insertados cumplan con el formato definido explicado en la HU 4.	

✂----------

Tarea	
Número tarea: 3	**Número y nombre historia:** HU 5: Gestionar datos de la competencia.
Nombre tarea: Validar entrada de datos.	
Tipo de tarea : Desarrollo	**Puntos estimados: 1**
Fecha inicio: 04/03/09	**Fecha fin:** 05/05/09
Programador responsable: Darién Concepción y Martha L. Gala	
Descripción: Todos los datos que son entrados son verificados y validados, en caso de que los datos sean entrados con errores el usuario será ratificado mediante un mensaje y se le brindará la posibilidad de insertar una vez más los datos correctamente. En caso de que los datos que fueron insertados ya existen en la BD implementada, se mostrará un mensaje de información al cliente.	

✂----------

Tarea	
Número tarea: 4	**Número y nombre historia:** HU 5: Gestionar datos de la competencia.
Nombre tarea: Eliminar datos.	
Tipo de tarea : Desarrollo	**Puntos estimados:** 1
Fecha inicio: 06/05/09	**Fecha fin**: 07/05/09
Programador responsable: Darién Concepción y Martha L. Gala	
Descripción:	

Se selecciona el dato del atleta que se quiere eliminar y al pulsar el botón eliminar, pues se pierde totalmente toda la información correspondiente de la BD.

✂----------

Tarea	
Número tarea: 1	**Número y nombre historia:** HU 6: Modalidad y títulos.
Nombre tarea: Generar Tabla.	
Tipo de tarea : Desarrollo	**Puntos estimados:** 3
Fecha inicio: 08/05/09	**Fecha fin**: 11/05/09
Programador responsable: Darién Concepción y Martha L. Gala	
Descripción: Se toma de la BD los datos correspondientes a la cantidad de participantes por cada modalidad y por sexos, además cuenta la cantidad de medallas obtenidas por cada modalidad (dígase oro, plata y bronce) y muestra los resultados en una tabla.	

✂----------

Tarea	
Número tarea: 1	**Número y nombre historia:** HU 7: Ranking por modalidad
Nombre tarea: Generar tabla.	
Tipo de tarea : Desarrollo	**Puntos estimados: 3**
Fecha inicio: 12/05/09	**Fecha fin**: 15/05/09
Programador responsable: Darién Concepción y Martha L. Gala	
Descripción: Se toman de la BD los datos que fueron generados por cada atleta durante la competencia en dependencia de la modalidad seleccionada por el usuario. Los datos son mostrados ordenadamente empezando por los mejores resultados.	

✂----------

Tarea	
Número tarea: 1	**Número y nombre historia:** HU 8: Ranking por país.
Nombre tarea: Generar tabla.	

Tipo de tarea : Desarrollo	**Puntos estimados: 1**
Fecha inicio: 25/05/09	**Fecha fin**: 28/05/09
Programador responsable: Darién Concepción y Martha L. Gala	
Descripción: Se toman de la BD los datos que fueron generados por cada atleta durante la competencia y en todas las modalidades, y en dependencia del país al que pertenezca cada atleta se va sumando para obtener el total de puntos acumulados. Estos puntos acumulados son mostrados ordenadamente empezando por los mejores resultados.	

✂----------

Tarea	
Número tarea: 1	**Número y nombre historia:** HU 9: autenticar usuario.
Nombre tarea: Lectura del login / password	
Tipo de tarea : Desarrollo	**Puntos estimados: 3**
Fecha inicio: 29/05/09	**Fecha fin:** 01/06/09
Programador responsable: Darién Concepción y Martha L. Gala	
Descripción: Se muestra una ventana emergente al iniciar la aplicación en la que se solicita el nombre de usuario (login) y la contraseña (password). La contraseña se muestra con asteriscos (*) mientras se escribe.	

✂----------

Tarea	
Número tarea: 2	**Número y nombre historia:** HU 9: autenticar usuario
Nombre tarea: Comprobación de la corrección del login / password	
Tipo de tarea : Desarrollo	**Puntos estimados: 3**
Fecha inicio: 02/06/09	**Fecha fin**: 05/06/09
Programador responsable: Darién Concepción y Martha L. Gala	

Descripción: Una vez introducidos el login y el password, y pulsado el botón “Aceptar”, se comprueban que existan en la base de datos. Los usuarios restringidos son “Estadístico” y “Comisión Aseguradora”, con unos determinados permisos de acceso a las funcionalidades de la aplicación.

✂----------

Tarea	
Número tarea: 3	**Número y nombre historia:** HU 9: autenticar usuario.
Nombre tarea: Mostrar sólo los menús correspondientes al usuario	
Tipo de tarea : Desarrollo	**Puntos estimados: 2**
Fecha inicio: 06/05/09	**Fecha fin:** 08/06/09
Programador responsable: Darién Concepción y Martha L. Gala	
Descripción: Una vez validada la corrección del usuario, se muestran sólo los menús de acceso a las partes de la aplicación que le corresponden al usuario, ocultando aquellos menús que no le correspondan.	

Las tarjetas CRC

El uso de las tarjetas C.R.C (Class, Responsabilities and Collaboration) permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica.

Las tarjetas C.R.C representan objetos; la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad.

Nombre de la Clase	
Responsabilidades (Atributos de la clase).	**Colaboradores** **(**Clases con las que se relaciona)

Tabla 2: Tarjeta CRC

He aquí las tarjetas CRC que se crearon:

Nombre de la clase: Cuentas	
Usuario CI Pass	Persona.

✂----------

Nombre de la clase: Persona	
CI Id_provincia_estado Id_pais Pnombre Snombre Papellido Sapellido Sexo	Atleta. Delegado. Persona _ evento. Oficial. Provincia _ estado. Cuenta.

✂----------

Nombre de la clase: País.	
Id_pais Nombre_pais	Evento. Provincia _ estado.

✂----------

Nombre de la clase: Delegado _ Evento	
Id_delegado_evento Id_evento id_persona	Evento. Delegado.

✂----------

Nombre de la clase: Delegado	
CI Papel_dentro_delegacion	Persona.

✂----------

Nombre de la clase: Atleta.	
CI	Marca _ personal _ distancia. Marca _ personal_ tiempo. Disciplina _ atleta _ evento. Persona. Atleta _ disciplina _ evento _ fase _

	ronda

✂----------

Nombre de la clase: Provincia _ estado.	
Id_provincia_estado Nombre_provincia_estado Id_pais	País. Evento _ provincia _ estado. Instalación _ deportiva. Persona.

✂----------

Nombre de la clase: Evento _ provincia _ estado.	
Id_evento Id_provincia_estado Id_pais	Evento. Provincia _ estado.

✂----------

Nombre de la clase: Evento.	
Id_evento Nombre_evento Fecha_realizacion Nivel_de_competicion Id_pais	Atleta _ disciplina _ evento. Cronograma. Persona _ evento. País. Evento _ provincia _ estado. Atleta _ disciplina _ evento _ fase _ ronda.

✂----------

Nombre de la clase: Disciplina _ atleta _ evento.	
Nombre_disciplina Id_evento CI lugar	Atleta. Disciplina. Evento.

✂----------

Nombre de la clase: Marca _ personal _ tiempo.	
CI Marca_personal_tiempo Nombre_disciplina	Atleta. Disciplina.

✂----------

Nombre de la clase: Marca _ personal _ distancia.

CI Marca_personal_distancia Nombre_disciplina	Atleta. Disciplina.

✂----------

Nombre de la clase: Ronda	
Numero_ronda	Atleta _ disciplina _ evento _ fase _ ronda

✂----------

Nombre de la clase: Atleta _ disciplina _ evento _ fase _ ronda	
Nombre_disciplina Id_evfento CI Numero_ronda Nombre_fase Marca_tiempo Marca_distancia	Fase. Ronda. Disciplina. Evento. Atleta.

✂----------

Nombre de la clase: Fase.	
Nombre_fase	Atleta _ disciplina _ evento _ fase _ ronda

✂----------

Nombre de la clase: Instalación _ deportiva.	
Id_instalacion_deportiva Id_provincia_estado Nombre_instalacion Id_pais	Disciplina _ instalación. Provincia _ estado.

✂----------

Nombre de la clase: Disciplina _ instalación.	
Id_instalacion_deportiva Id_provincia_estado Id_pais Nombre_disciplina	Instalación deportiva. Disciplina.

✂----------

Nombre de la clase: Cronograma.	
Id_evento Direccion.	Evento.

✂----------

Nombre de la clase: Disciplina.	
Id_disciplina Nombre_disciplina Record_olimpico Record_mundial Atleta_ro Atleta_Rm Tipo	Especialidad. Record. Oficial _ disciplina. Disciplina _ Instalación. Disciplina _ atleta _ evento. Marca _ personal _ tiempo. Marca _ personal _ distancia. Atleta _ disciplina _ evento _ fase _ ronda.

✂----------

Nombre de la clase: Oficial.	
Function CI	Persona. Oficial _ disciplina.

✂----------

Nombre de la clase: Oficial _ Disciplina.	
CI Nombre_disciplina	Disciplina. Oficial.

✂----------

Nombre de la clase: Record.	
Id_record Nombre_disciplina Nombre_recordista Nombre_evento Tipo_record Fecha_record Marca_distancia Marca_tiempo	Disciplina.

✂----------

Nombre de la clase: Especialidad.	
Nombre _ especialidad	Disciplina.

Pruebas de Aceptación.

Crear test que prueben el funcionamiento de los distintos códigos implementados ayudará a desarrollar dicho código. Crearlos antes ayuda a saber qué es exactamente lo que tiene que hacer el código a implementar y teniendo la certeza que una vez implementado pasará dichos test sin problemas ya que dicho código ha sido diseñado para ese fin.

Test de aceptación.

Para asegurar el funcionamiento final de una determinada HU se debe crear un "Test de aceptación"; este es creado y usado por el cliente para comprobar que las distintas HU cumplen su cometido.

Estas pruebas se realizaron para tener un conocimiento de las funcionalidades que debía cumplir el sistema y fueron redactadas en conjunto con el representante del cliente. Aunque el equipo de desarrollo determinó llamarles funcional a este tipo de prueba, son en realidad las pruebas de aceptación por las que deberá regirse la implementación de cada HU que sea puesta en el repositorio para ser refactorizada o consultada para posteriores implementaciones, debido a que estas son las pruebas por las que el representante del cliente verificará si el sistema hace o no lo que él desea.

Para el desarrollo de la misma se utiliza una plantilla estándar de casos de pruebas la cual se expone de la siguiente manera:

Descripción General

[Descripción de Aspectos Generales a tener en cuenta a la hora de realizar el diseño de las pruebas, incidencias en el momento de su desarrollo y otros aspectos relevantes.]

A esta Historia de Usuario se le realizaron las siguientes pruebas:

[Lista de las Pruebas diseñadas para ser aplicadas a la Historia de Usuario, así como las funcionalidades para las que se diseñaron las mismas.]

CPR 1: <Funcionalidad #1 a revisar>

Descripción de la Funcionalidad:

[Descripción de la Funcionalidad a Probar].

Flujo Central:

[Flujo Central del Caso de Prueba. Pasos a desarrollar para probar la Funcionalidad que se indicó].

Condiciones de Ejecución:

[Prerrequisitos imprescindibles para que se pueda ejecutar correctamente el Caso de Prueba].

Iteraciones:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
<Clases Válidas>	<Clases Inválidas>	<Resultado Esperado de la Prueba según lo Especificado en la Historia de Usuario>	<Resultado Real de la Prueba>	<Observaciones>

Registro de defectos y dificultades detectados.

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código de CP	Importancia	Recomendación
<Nombre del Elemento>	<1>	<Descripción de la No Conformidad>	<Descripción de Aspecto correspondiente>	<Etapa de detección del error>	<X>	<X>

De esta manera quedan conformadas las pruebas de aceptación para todas las HU en orden con la iteración a la que pertenecen.

Iteración 1.

1. **HU Asegurar evento.**

En esta HU La Comisión Aseguradora tiene la opción de introducir los datos necesarios para realizar las operaciones deseadas las cuales se recogen en las HU posteriores, las cuales son detalladas en su momento.
La Comisión Aseguradora es la máxima responsable del buen desarrollo del evento en cuestión. Es por eso que mediante esta HU se le facilita su trabajo pues la misma le da la posibilidad de realizar distintas operaciones como son la gestión de los datos personales de los participantes en el evento o sea los deportistas.
Esto le permite mejor organización en sentido general además estos datos serán guardados en la base de datos que ha sido creada para el submódulo en cuestión. La cual tendrá siempre la información correspondiente a los eventos a realizar facilitando el trabajo ya que solo bastará hacer una consulta a la misma para obtener eficientemente los datos que se requieren saber.

Además debe identificarse con un logo que promocione las principales características del evento que se esta realizando.

A esta Historia de Usuario se le realizaron las siguientes pruebas:
Funcional.

CPR 1: Selección de las distintas acciones a realizar.
CPR 2: Insertar correctamente los datos.

Descripción de la Funcionalidad:
El encargado de la comisión aseguradora del evento debe autenticarse para poder realizar cualquier operación, correspondiente a la creación de las características propias del evento, la gestión de los datos personales de los participantes del evento.

Una vez autenticado puede seleccionar cualquier opción disponible y realizar la gestión de los datos, según la opción seleccionada.

Flujo Central:
Opción autenticar usuario/usuario/contraseña.
Opción inscripción de atletas/introducir datos.

Opción Asegurar Evento/insertar los datos del evento.

Condiciones de Ejecución:

El encargado de la comisión aseguradora deberá estar registrado y autenticado.

Iteraciones:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Autenticación del usuario.		El usuario podrá realizar cualquier acción que desee, ya que tendrá privilegios para realizar cualquier actividad.	Satisfactoria.	
	Usuario no autentica-do.	El usuario no podrá realizar cualquier acción que desee, ya que no tendrá privilegios para realizar dicha acción. Sólo podrá ver los datos que estén registrados mediante la opción mostrar.	Satisfactoria.	
Selecciona asegurar evento.		Se muestra una interfaz que contiene los campos a llenar.	Satisfactoria.	
Selecciona Gestionar datos de los atletas.		Se muestra una interfaz que permite la inserción de los datos correspondientes. (Ver HU gestionar datos de los atletas.)	Satisfactoria.	

Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Autenticar usuario.	1	Datos incorrectos	Los datos que fueron insertados no son correctos o no corresponden el uno con el otro. (o sea el nombre del usuario no corresponde con la contraseña)	Implementación.	alta	Validar datos.
Asegurar evento.	2	Datos incorrectos	Los datos insertados son incorrectos, o faltan campos por llenar.		media	Validar datos. Mostrar mensaje de verificación.

Seleccionar la opción deseada.	3	Mostrar interfaces correspon dientes.	Mostrar correctamente las interfaces correspondientes (ver casos de pruebas correspondientes a estas opciones).	Implementación.	media	Verificar conexión.

2. HU Gestionar Evento.

En la HU gestionar evento el encargado de la comisión organizadora tiene la posibilidad de modificar, eliminar y mostrar el listado que contienen la información correspondiente al evento o sea nombre de del evento, año que se celebra, logo del mismo. Este listado estará listo en la BD para ser seleccionado y realizar cualquier acción que se desee con la información que contiene.

A esta Historia de Usuario se le realizaron las siguientes pruebas:
Funcional.

CPR 1: Correcta seleccion del evento.
CPR 2: Modificar Evento.
CPR 3: Eliminar Evento.
CPR 4: Mostrar Evento.

Descripción de la Funcionalidad:

El encargado de la comisión organizadora del evento debe seleccionar la opción de modificar, eliminar, o mostrar Evento. Una vez seleccionada la opción que se desea realizar se podrá realizar esta acción solo con especificar el nombre del evento.

Flujo Central:

El encargando selecciona la opción Evento.
Evento/modificar evento.
Evento/eliminar evento.
Evento/mostrar evento.

Condiciones de Ejecución:

El encargado de la comisión aseguradora deberá estar registrado y autenticado.

Iteraciones:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Selecciona la opción modificar evento.		Se mostrará una nueva interfáz que al seleccionar el evento que se desea modificar, mostrando los datos correspondientes y permitiendo hacer los cambios pertinentes.	Satisfactoria.	Los campos serán habilitados para posteriores cambios.
	Selecciona otra opción.	Se muestra la interfaz correspondiente a dicha	Satisfactoria.	

		opción.		
Selecciona la opción eliminar Evento.		Se muestra una interfaz que permite seleccionar el Evento y la opción eliminar. Se eliminará completamente de la BD.	Satisfactoria.	Mostrar un mensaje de confirmación.
Selecciona la opción mostrar Datos del Evento.		Se muestra una interfaz que permite seleccionar el evento que se desea visualizar, y se mostrarán todos los datos del mismo.	Satisfactoria.	

Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Interfaces incorrectas.	1	Interfaz no deseada	Se mostro una interfaz que no es correspondiente a la opción seleccionada.	prueba	media	Verificar conexión de interfaces.
Desactualizada la BD.	2	La BD no se actualiza luego de una operación.	La BD muestra datos incorrectos pues no se actualiza debidamente después de realizar cualquier operación.	prueba	media	Revisar funcionalidadese implementacio-nes de la BD.

3. HU Gestionar Inscripción de Atletas.

En la HU gestionar inscripción de atletas se le brinda la posibilidad al encargado de la comisión organizadora de insertar, modificar, eliminar y mostrar todos los datos del atleta. Datos como nombre completo, modalidad, número de carnet de identidad, grupo sanguíneo, nacionalidad, marcas personales, en la temporada, record olímpico, record mundial.

A esta Historia de Usuario se le realizaron las siguientes pruebas:
Funcional.

CPR 1: Correcta inserción de Datos
CPR 2: Modificar Datos.
CPR 3: Eliminar Datos
CPR 4: Mostrar Datos.

Descripción de la Funcionalidad:
El encargado de la comisión aseguradora deberá seleccionar la opción gestionar datos del atleta y se mostrará una nueva interfáz donde podrá realizar cualquier acción que desee con los datos del atleta.

En la funcionalidad de insertar los datos el responsable podrá insertar todos los datos relacionados con cada atleta, datos como nombre completo, modalidad, número de carnet de identidad, grupo sanguíneo, nacionalidad, marcas personales, en la temporada, record olímpico, record mundial.
En la funcionalidad modificar se brinda la posibilidad de introducir el nombre de un atleta y si este existe en la BD entonces mostrará los campos que puedan ser modificados. Guardando así los cambios en la BD.
En la opción eliminar con solo introducir el nombre del atleta se obtendrá como resultado un listado con todos sus datos y la opción de si desea o no eliminarlo. De ser así se eliminara completamente de la BD.
En la opción mostrar se visualiza un listado con todos los nombres de los atleta, dando la posibilidad de introducir el nombre de cualquiera que se desee y mostrará un listado con todos sus datos personales.

Flujo Central:

Seleccionar la opción gestionar atleta.
Atleta/insertar datos.
Atleta /modificar datos.
Atleta /eliminar participante.
Atleta / mostrar datos.

Condiciones de Ejecución:

El encargado de la comisión deberá estar registrado y con los privilegios pertinentes.

Iteraciones:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Selecciona la opción gestionar inscripción del Atleta.		Se mostrará una interfaz que permite realizar las distintas funcionalidades correspondientes a los datos del mismo.	Satisfactoria.	Al realizar cualquier operación se realizará una verificación de existencia en la BD, para luego realizar cualquier operación. Luego de realizarla se debe enviar un mensaje de confirmación.
	Selecciona otra opción.	Muestra la interfaz correspondiente a la opción seleccionada.		
Selecciona insertar datos.		Muestra una interfaz que permite insertar todos los datos necesarios de cada Atleta.	Satisfactoria.	Todos los datos son guardados en la BD.

Selecciona eliminar datos.		Visualiza una interfaz que permite insertar el nombre del Atleta, luego muestra sus datos y la opción eliminar.	Satisfactoria.	Borrando así todos los datos de la BD.
Selecciona modificar datos.		Visualiza una interfaz que permite insertar el nombre del Atleta que se desea modificar, luego se muestra un listado con sus datos personales mostrando activos los campos pertinentes para ser cambiados.	Satisfactoria.	Todos los cambios serán registrados en la BD.
Selecciona mostrar datos.		Muestra un listado con todos los nombres de los Atleta, y da la posibilidad de insertar el nombre del que desee conocer más, luego de esta acción mostrará un listado con todos los datos personales del mismo.	Satisfactoria.	Estos listados son cargados de la BD.

Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Inserción incorrecta de datos.	1	Los datos son incorrectos.	La inserción de los datos en los distintos campos no son validos.	prueba	Media.	Validar entrada de datos.

Iteración 2.

4. HU Archivar Datos de la Competencia.

En esta HU el estadístico tiene la posibilidad de insertar todos los datos que son generados durante la ejecución del evento y al terminar cada modalidad. Estos datos son

los que obtiene cada atleta al realizar su modalidad, o sea cada corredor, saltador, o lanzador al realizar su evento o sea la modalidad en la que esta inscrito registra una nueva marca, esta marca será guardada en el sistema o sea la BD con sus unidades de medida correspondientes, para luego ser consultada y realizar cualquier operación que se desee con ella.

A esta Historia de Usuario se le realizaron las siguientes pruebas: Funcional.

CPR 1: Insertar correctamente los datos.

Descripción de la Funcionalidad:
El estadístico deberá seleccionar correctamente la modalidad que se va a realizar para así poder insertar los datos que sean generados por cada atleta. Luego se mostrará un listado con los nombres de los atletas y un campo disponible para registrar el dato generado. Al seleccionar la opción aceptar o guardar todos estos cambios serán guardados en la BD.

Flujo Central:
Selecciona Archivar datos.
Archivar datos/modalidad.
Archivar datos/insertar datos.

Condiciones de Ejecución:
El estadístico deberá estar registrado y con los privilegios pertinentes para realizar dicha operación.

Iteraciones:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Selecciona archivar datos.		Se visualiza una interfaz que permite seleccionar la modalidad que se desea para hacer el registro pertinente.	Satisfactoria.	
	Selecciona otra opción.	Muestra la interfaz correspondiente a la opción seleccionada.	Satisfactoria.	
Selecciona modalidad.		Muestra un listado con los nombres de cada atleta que fue inscrito en esa modalidad y un campo disponible para la inserción de datos.	Satisfactoria.	Al seleccionar la opción aceptar guardara todos los cambios en la BD. Y mostrará un mensaje de ratificación.

Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Inserción de datos	1	Los datos son incorrectos.	Los datos que fueron insertados son incorrectos. Solo se permiten números y puntos.	implementación.	Media.	Validación de los datos que son insertados. Enviar un mensaje de verificación.

Iteración 3

5. HU Gestionar Datos de la Competencia.

En esta HU el estadístico podrá realizar cualquier operación que desee con los datos que fueron registrados con anterioridad. Estas acciones pueden ser modificar, eliminar, o mostrar los datos que obtuvieron como resultado los deportista en su modalidad, todos estos cambios será guardado en la BD como constancia de las acciones realizadas.

A esta Historia de Usuario se le realizaron las siguientes pruebas:
Funcional.

CPR 1: Eliminar Datos de la Competencia
CPR 2: Modificar Datos de la Competencia.
CPR 3: Mostrar Datos de la Competencia.

Descripción de la Funcionalidad:
El estadístico tiene la posibilidad de seleccionar que opción desea realizar con respecto a los datos que fueron generados y archivados en la BD.
Si elije la opción modificar datos deberá insertar el nombre del atleta mostrándose así un listado con los datos de este atleta y los campos disponibles para el cambio que se desea hacer, estos cambios serán guardados automáticamente en la BD.
Si elije la opción eliminar datos debe insertar el nombre del deportista, luego se mostrara un listado con los datos de este deportista, con un botón que da la posibilidad de eliminar todo lo referente a los datos generados por ese deportista. Si se elimina todo lo referente a esos datos desaparecerá totalmente de la BD.
Si elije la opción mostrar se visualizará una interfáz con un listado que contiene los nombres de los deportistas y los datos correspondientes a cada uno. Si lo que deseas es consultar los datos de un deportista en particular pues solo debe insertar el nombre del deportista en el campo que estará disponible y al aceptar se mostrará un nuevo listado con todos los datos del deportista seleccionado.

Flujo Central:
Selecciona modificar datos.
Selecciona eliminar datos.
Selecciona mostrar datos.

Condiciones de Ejecución:

El estadístico debe estar registrado y con los privilegios pertinentes con anterioridad.

Iteraciones:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Selecciona modificar datos.		Muestra una interfaz donde se debe insertar los datos del deportista que se desee y este mostrara un listado con los datos del mismo y los campos pertinentes a disposición para ser modificados.	Satisfactoria.	Se hace una consulta en la BD y si existe se muestran los datos del mismo y luego se cambian los datos que fueron modificados. Se envía un mensaje de notificación.
	Selecciona otra opción.	Mostrara la interfaz solicitada.	Satisfactoria.	
Seleccion a eliminar datos.		Muestra una interfaz donde se debe insertar los datos del deportista que se desee y este mostrará un listado con los datos del mismo y los campos pertinentes a disposición para ser eliminados.	Satisfactoria.	Se hace una consulta en la BD y si existe se muestran los datos del mismo y luego eliminan los datos que se creyeron pertinentes de la BD. Se envía un mensaje de notificación.
Seleccion a mostrar datos.		Visualiza una lista con los nombres y datos registrados de cada deportista. Con una opción que le permite insertar el nombre de un deportista y se mostrará un listado con los datos correspondientes al mismo.	Satisfactoria.	Estos listados son generados de la BD.

Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Inserción de nombre.	1	Existencia del nombre en la BD.	El nombre que fue insertado no existe en la BD.	prueba.	alta.	Verificar la BD en caso de que no exista ese nombre mostrar un mensaje de notificación.

Inserción de datos.	2	Datos incorrectos.	Los datos entrados son incorrectos solo se permiten números y puntos.	prueba.	media.	Validar datos. Enviar un mensaje de verificación.

6. Modalidad y Titulos.

Esta HU brinda la posibilidad a la comisión aseguradora del evento obtener mediante una consulta a la BD obtener la relación de participación por modalidades y sexo, así como la relación de las medallas por modalidades.

A esta Historia de Usuario se le realizaron las siguientes pruebas:
Funcional.

CPR 1: Mostar listado

Descripción de la Funcionalidad:
Al seleccionar la opción modalidad y títulos se visualizará una tabla que muestra la modalidad con la cantidad de participantes por sexo, así como un total de los mismos. Y otra división relacionada con los títulos o sea con las medallas mostrando la cantidad de cada una y una suma del total de ellas. Quedaría así aproximadamente:

	Sexo				Medallas			
Modalidad	F	M	Total		Oro	Plata	Bronce	Total
4 x 400	X	x	xx		x	x	x	Xx
Jabalina	x	x	xx		x	x	x	Xx
Salto alto	x	x	xx		x	x	x	xx

Flujo Central:
Selecciona modalidad y títulos.

Condiciones de Ejecución:
Debe estar registrado en el sistema además debe haber ocurrido ya el evento.

Iteraciones:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Selecciona modalidad y títulos.		Visualiza un listado todos los datos referentes a la participación por sexo en cada modalidad, y la cantidad de medallas.	Satisfactoria.	Estos datos son generados desde la BD.
	Selecciona otra opción.	Muestra la interfaz correspondiente.	Satisfactoria.	

Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Inexistencia de los datos.	1	Muestra datos que no han sido creados.	Muestra datos que no han sido generados con anterioridad, o en ocasiones datos que no corresponden con la realidad.	prueba.	media.	Verificar la consulta a la BD, y la entrada de datos.

7. HU Ranking X Modalidad.

Esta HU genera mediante una consulta a la BD el ranking por modalidades para los atletas.

A esta Historia de Usuario se le realizaron las siguientes pruebas:
Funcional.

CPR 1: Seleccionar Modalidad.
CPR 2: Mostrar listado.

Descripción de la Funcionalidad:
Selecciona la opción ranking por modalidad, luego en la nueva interfáz se elige la modalidad y se muestra un listado de todos los atletas que participaron en esa modalidad, organizados de mayor a menor en los resultados.

Flujo Central:
Selecciona Ranking X modalidad.
Selecciona modalidad.

Condiciones de Ejecución:
Debe existir con anterioridad datos en la BD para poder tener éxitos en la consulta.

Iteraciones:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Selecciona Ranking X modalidad.		Muestra una interfaz que permite seleccionar la modalidad que se desea conocer.	Satisfactoria.	
	Selecciona otra opción.	Muestra la interfaz correspondiente	Satisfactoria.	
Selecciona la modalidad.		Muestra un listado con los nombres de los atletas y los datos	Satisfactoria.	Este listado es ordenado de mayor a menor correspondiente a

		correspondientes.		los resultados registrados.

Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendació
Datos inexistentes.	1	Muestra datos que no existen, o datos no correspondientes	Muestra datos que son equívocos, o datos no reales.	prueba.	media.	Verificar consulta a la BD. Enviar mensaje de verificación.

Iteración 4

8. HU Ranking X Países.

En esta HU genera a partir de una consulta a la BD del sistema una tabla que contiene los nombres de cada país participante y los puntos que obtuvieron durante la ejecución del evento en cuestión.

A esta Historia de Usuario se le realizaron las siguientes pruebas:
Funcional.

CPR 1: Mostrar tabla con datos.

Descripción de la Funcionalidad:
Selecciona la opción Ranking por países, luego en la nueva interfáz se muestra un listado de todos los países que participaron en ese evento así como la puntuación que obtuvieron, organizados de mayor a menor en los resultados.

Flujo Central:
Selecciona Ranking X Países.

Condiciones de Ejecución:
El usuario debe estar registrado, además deben existir datos en la BD correspondientes al evento en cuestión.

Iteraciones:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Selecciona Ranking X países.		Muestra una interfaz con una tabla donde muestra los países con los datos correspondientes al evento realizado.	Satisfactoria.	Se hace una consulta en la BD se muestran los datos correspondientes.
	Selecciona otra opción.	Mostrará la interfaz solicitada.	Satisfactoria.	

Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Datos inexistentes.	1	Existencia de los datos en la BD	No existen datos en la BD.	prueba.	alta.	Verificar la BD en caso de que no existan datos registrados, mostrar un mensaje de notificación.
Datos no correspon-dientes.	2	Datos incorrectos.	Los datos mostrados son incorrectos ya que no coinciden con la opción seleccionada.	prueba.	alta.	Verificar la consulta a la BD.

9. Autenticar Usuario.

Esta HU brinda gran seguridad para la comisión aseguradora, pues al implementarla permitirá que solo los usuarios autenticados puedan trabajar en la aplicación según los roles que se le hallan definido con anterioridad.

A esta Historia de Usuario se le realizaron las siguientes pruebas:
Funcional.

CPR 1: Insertar los Datos del usuario

Descripción de la Funcionalidad:
La primera acción que debe realizar el usuario antes de entrar al sistema o de intentar realizar cualquier opción será registrase con su nombre de usuario y contraseña. Solo así podrá realizar las operaciones que desee y según se les permita, siguiendo el criterio de los roles determinados para cada tipo de usuario.

Flujo Central:
Registrar usuario.

Condiciones de Ejecución:
Es la primera interfáz que se muestra.

Iteraciones:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Insertar datos del usuario.		Visualiza la interfaz principal y permite realizar las actividades según los privilegios.	Satisfactoria.	

	Selecciona otra opción.	No permite realizar ninguna operación hasta tanto no este registrado.	Satisfactoria.	

Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Inexistencia de los datos.	1	El usuario no esta registrado en la BD.	Los datos que fueron entrados no se encuentran en la BD registrados.	prueba.	alta.	Verificar la consulta a la BD, y la entrada de datos. Enviar mensaje de notificación.
Datos incorrectos.	2	Los datos entrados son incorrectos.	Los datos que fueron insertados por el usuario son incorrectos pues no coinciden unos con otros.	prueba.	alta.	Realizar una consulta a la BD y mandar un mensaje de verificación.

10. HU Gestionar Usuario.

Esta HU permite insertar, modificar o eliminar los roles de un usuario que halla sido registrado en la BD.

A esta Historia de Usuario se le realizaron las siguientes pruebas:
Funcional.

CPR 1: Insertar Datos del usuario.
CPR 2: Modificar Datos del Usuario.
CPR 3: Eliminar Datos del Usuario.

Descripción de la Funcionalidad:
El administrador tiene la potestad de crear, modificar o eliminar un usuario así como darle los roles pertinentes para que realicen cada operación debidamente.
Si elige la opción de crear un nuevo usuario pues podrá crearlo debidamente o sea con nombre y contraseña, así como el rol que desempeñara el cual delimitará el rango de operaciones que pueda realizar.
Si elige modificar, pues podrá modificar el rol que desempeña delimitando bien el rango de responsabilidades u operaciones que pueda realizar.
Si elige eliminar un usuario este desaparecerá totalmente de la BD.

Flujo Central:
Selecciona crear usuario.
Selecciona modificar usuario.
Selecciona eliminar usuario.

Condiciones de Ejecución:
El administrador es el único que puede realizar estas operaciones.

Iteraciones:

Clases Válidas	Clases Inválidas	Resultado Esperado	Resultado de la Prueba	Observaciones
Selecciona crear usuario.		Inserta en la BD los datos necesarios del nuevo usuario, así como define el rol que desempeñara.	Satisfactorio.	Lo hace directamente a la BD.
	Selecciona otra opción.	Muestra los campos correspondiente	Satisfactorio.	
Selecciona modificar usuario.		Puede modificar el rol del usuario asignándole o no un nuevo rol.	Satisfactorio.	
Selecciona eliminar usuario.		Elimina el usuario y se pierde totalmente los datos referentes al mismo de la BD.	Satisfactorio.	

Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección Código del CP	Importancia	Recomendación
Datos inexisten-tes.	1	Muestra datos que no existen, o datos no correspondientes.	Muestra datos que son equívocos, o datos no reales.	prueba.	media.	Verificar consulta a la BD. Enviar mensaje de verificación.

Conclusiones.

En este capítulo se vieron temas como las tarjetas CRC, las tareas de ingenieria y las pruebas de aceptación, todos ellos con el objetivo de lograr un diseño sencillo y óptimo o sea lograr un sistema que cumpla con las necesidades del cliente, fácil de implementar y de entender, y que el usuario del sistema no necesite de mucha preparación para el entendimiento del mismo.

Conclusiones generales.

Luego de haber desarrollado el presente trabajo diploma que además de constituir el último trabajo como estudiantes de La Universidad de las Ciencias Informáticas, y del cumplimiento de los principales objetivos que conllevaron a la realización del mismo, se han abierto las puertas de la era de digitalización al deporte pues con la aplicación del sistema obtenido se dará gran ayuda a aquellos que desarrollan su trabajo en el Inder Nacional.

Para la realización de un buen sistema que cumpla con todas las necesidades y expectativas del cliente, se desarrolló una profunda investigación con el objetivo de lograr una buena planificación para el equipo de desarrollo y conocer sobre las tendencias más actuales en cuanto a tecnología, para así poder decidir las herramientas más apropiadas según las condiciones del cliente.

El sistema implemetado en el lenguaje Java permite obtener una aplicación de escritorio con el objetivo de mantener la seguridad de los datos, asegurándose así que aunque no exista el acceso a ellos mediante la red se puedan seguir guardando en la Base de Datos desarrollada en el PostgreSQL; logrando de esta manera que al acceder a las estadísticas generadas por los datos obtenidos luego del desempeño de cada atleta, estén siempre disponibles y a salvo. Esta aplicación es sencilla y amigable pues se desarrolló siguiendo las políticas y principios que plantea la metodología de desarrollo conocida como eXtreme Programming (XP). Se creó además la documentación necesaria y pertinente, que son propuestas por la metodología utilizada, para que nuevos integrantes del equipo de desarrollo puedan entender las bases sobre las cuales se desarrolló este sistema.

Con esta aplicación se facilita el desempeño de los trabajadores del Inder, especialmente los que pertenecen a la Asociación del Atletismo en Cuba, pues con el mismo se agilisa cualquier operación que se realiza manualmente. Estas operaciones pueden ser por ejemplo la inscripción de los participantes del evento, y la gestión de las estadísticas obtenidas en el evento, dígase entre otras cosas la inserción de los datos generados durante la competencia por cada atleta en las distintas modalidades.

Recomendaciones.

Durante la realización de esta investigación surgieron ideas que pueden servir como recomendación para el perfeccionamiento del sistema, ellas son:

Seguir adelante con esta investigación y se extienda a otros deportes ya que su resultado es beneficioso para el desarrollo informático del país.

Realizar mantenimiento y actualizaciones al sistema obtenido como resultado de nuestra investigación, adicionándole nuevos módulos y funcionalidades que beneficien los resultados obtenidos, en dependencia del avance tecnológico del país.

Realizar refactorización del software y del código, con el objetivo de mejorar el sistema, pues con dicha refactorización (o sea, reescribir ciertas partes del código para aumentar su legibilidad pero sin modificar su comportamiento), se podrán usar los códigos ya implementados con el objetivo de eliminar aquellos que contienen funcionalidades que no serán usadas, ya que pueden generar código completamente inestable y mal diseñado.

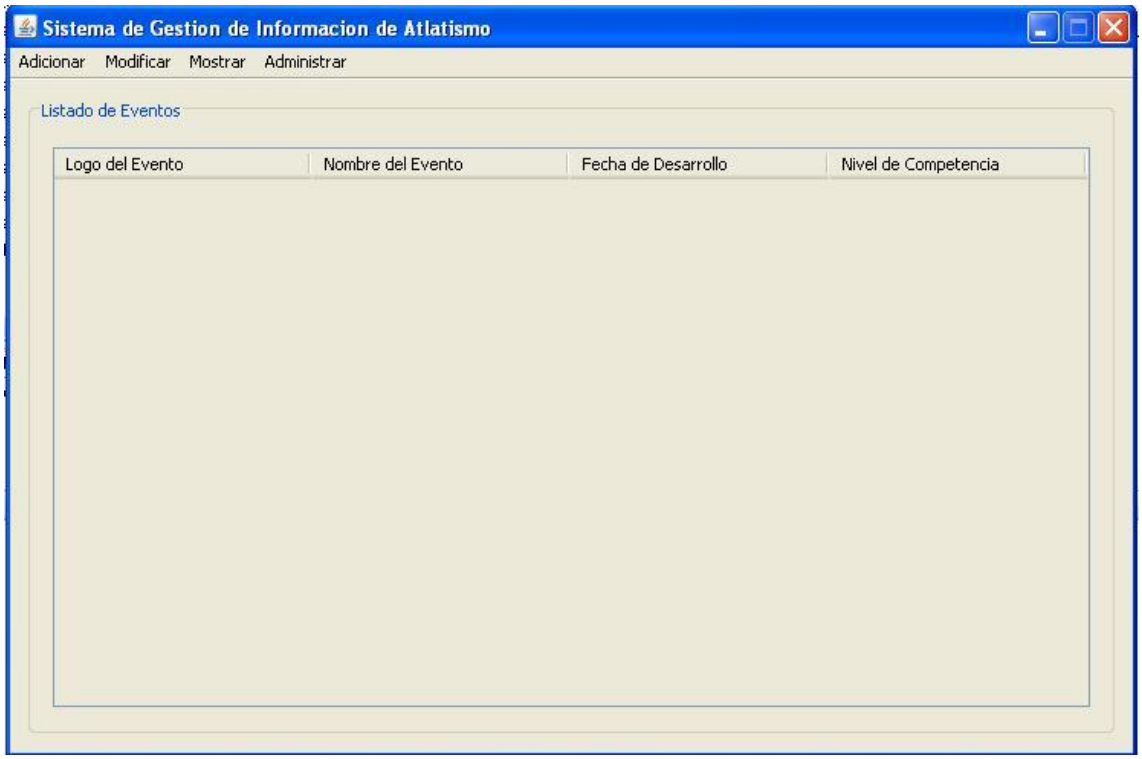
Anexos:

Anexo 1: Interfaz Inicial.

Esta es la interfaz que se mostrará al iniciar el sistema. Para acceder a la interfaz principal deberá registrarse, de lo contrario no podrá acceder. Una vez registrado podrá realizar cualquier operación que desee.

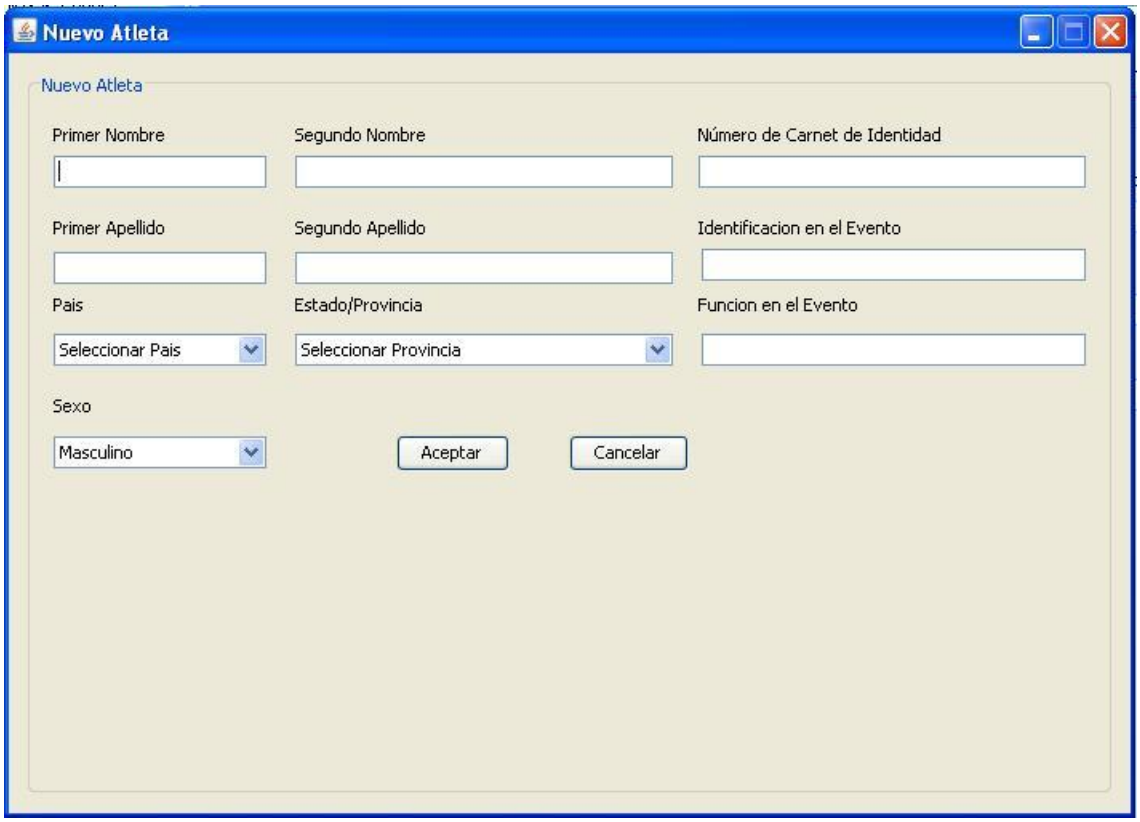
Anexo 2: Interfaz Principal.

Esta es la interfaz principal del sistema de la cual se deriban las demás, para hacer cumplir las necesidades del cliente y las principales funcionalidades del mismo.



Anexo 3: Interfaz Adicionar Atleta

Esta interfaz es una de las deribadas de la vista principal, mediante la cual se insertan los datos de todos los atletas participantes.



Anexo 4: Tareas de ingeniería para mejor comprensión.

Todas las modalidades realizarán distintas etapas que son conocidas como clasificatoria, eliminatorias, y finales. Y por supuesto cada una tiene su forma característica de hacerlo, y se debe al tipo de especialidad que sea si es de tiempo o de distancia, por lo que la dividimos las tareas en dos momentos, o sea primero la especialidad de pista que corresponde con el tiempo y luego las especialidades de campo y lanzamiento de artefactos que responden a la distancia. Para mejor entendimiento se desglosarán según como se desarrollen en cada especialidad y se plasmarán en las tareas de ingeniería las cuales pertenecen a la HU Archivar Datos de la Competencia. Estas tareas quedaran explicitas de la siguiente manera.

Especialidad de Pista.

Las distintas series o sea la cantidad de veces que se ejecuta esta modalidad se determina a partir de la cantidad de participantes en una modalidad determinada. Aunque esto forma parte del módulo organizativo es importante que el estadístico lo sepa ya que de esta manera sabrá que atleta está en cada carril y en cada fase ya sea clasificatoria, eliminatoria, o final. Para la determinación de los participantes a la próxima fase es casi evidente ya que seguirán adelante aquellos atletas que pasen por la línea de meta cualquier parte de su cuerpo, obteniendo así su tiempo de ejecución, el cual le dará o no la posibilidad de pasar a la próxima fase.

Es valido señalar que estas tareas de ingeniería son validas para todas las especialidades de pista, o sea carreras en todas sus distancias, carrera con obstáculos, y carrera de relevos.

	Tarea
Número tarea: 4	**Número y nombre historia:** HU 4 archivar datos de la competencia.
Nombre tarea: Fase clasificatoria y eliminatoria**.**	
Tipo de tarea : Desarrollo	**Puntos estimados: 1**
Fecha inicio:	**Fecha fin**:
Programador responsable:	
Descripción: Una vez el atleta pase cualquier parte de su cuerpo se tomará el tiempo y se insertará en el sistema propuesto como solución, al terminar con todos los atletas de esa eliminatoria previa y haber guardado los resultados, pues este deberá organizar los tiempos obtenidos comenzando por los mejores. De esta manera se podrá determinar con mejor facilidad los participantes a la próxima fase.	

✂----------

	Tarea
Número tarea: 5	**Número y nombre historia:** HU 4 archivar datos de la competencia.
Nombre tarea: Fase final.	
Tipo de tarea : Desarrollo	**Puntos estimados: 1**
Fecha inicio:	**Fecha fin**:
Programador responsable:	
Descripción: En la final como su nombre lo indica estarán aquellos atletas que obtengan los mejores tiempos, una vez organizados en los carriles correspondientes y dada la señal de salida solo se esperará a que el atleta pase por la línea de meta cualquier parte de su cuerpo obteniendo así su tiempo el cual le permitirá ser o no el ganador de esta modalidad.	

Los empates se resolverán como sigue:

Al determinar si hubo empate en cualquier eliminatoria para el puesto que permite el paso a la siguiente fase, basado en el tiempo realizado, el Juez Jefe de Foto-Finish deberá

considerar los tiempos registrados en 1/1000 de segundo. Si se determina que hubo empate, los atletas empatados pasarán a la siguiente eliminatoria o, si esto no fuera posible, se hará un sorteo para determinar quién será calificado para la eliminatoria siguiente. En caso de empate para el primer puesto en una final, el Juez Árbitro está autorizado para decidir, si es posible, que los atletas empatados compitan de nuevo. Si se decide que no, el resultado subsistirá. Los empates para otros puestos serán válidos.

En las **especialidades de campo** que son las dedicadas a los saltos, así como el **lanzamiento de artefactos** se realizan también las distintas fases correspondientes o sea clasificatorias, eliminatorias y finales, solo que se realizan de forma distinta, o sea a partir de intentos válidos, fallidos, o nulos. Como en ambas especialidades el procedimiento para cada fase se hace por intentos, pues se explica de forma general y por especialidades como se cuentan los intentos válidos o no los cuales son el pilar más importante para la toma de decisiones en cada caso. Y el procedimiento es el siguiente:

Especialidades de campo.

Salto verticales.

Antes de comenzar la competición el Juez Jefe dará a conocer a los atletas la altura a que se colocará el listón al iniciarse la prueba, y las distintas alturas a que será elevado al final de cada vuelta hasta que solamente quede un atleta ganador de la competición o haya un empate para el primer puesto.

Un atleta puede comenzar a saltar a cualquier altura de las previamente anunciadas por el Juez Jefe de la prueba y puede continuar saltando a su discreción a cualquier altura superior a aquella. Después de tres intentos nulos consecutivos, cualquiera que fuera la altura en que se hubiesen producido, el atleta no puede seguir efectuando más saltos, excepto en el caso de un empate para el primer puesto. El atleta puede renunciar a su segundo o tercer intento en una altura determinada eligiendo así una mayor altura y realizar los intentos correspondientes.

Saltos horizontales.

En las pruebas de saltos horizontales las distancias deben registrarse hasta el 0,01 m. inferior a la distancia medida, si ésta no es un centímetro entero.

Lanzamiento de artefactos.

En los lanzamientos de peso, disco y martillo los artefactos deberán ser lanzados desde un círculo y la jabalina desde un pasillo. En el caso de los intentos hechos desde un círculo, el atleta tiene que comenzar desde una posición estacionaria dentro del círculo. Se permite al atleta tocar el interior del aro que forma el círculo. En el lanzamiento de peso también podrá tocar el interior del contenedor.

El lanzamiento será considerado como nulo si el atleta en el curso de un intento:

a) Suelta indebidamente el peso o la jabalina.
b) Después de haber penetrado en el interior del círculo e iniciado un lanzamiento toca con cualquier parte de su cuerpo la parte superior del anillo de hierro o el suelo del exterior del círculo.

c) En el lanzamiento de peso toca con cualquier parte de su cuerpo la parte superior del contenedor.
d) En el lanzamiento de jabalina toca con cualquier parte de su cuerpo las líneas que delimitan el pasillo o el terreno exterior.

Nota: El lanzamiento no será considerado como nulo si el disco o cualquier parte del martillo choca con la jaula después de soltarlo sin que ningún otro Artículo sea infringido.

Nota: Todos los movimientos permitidos por este apartado deberán realizarse en el tiempo máximo concedido a un intento

Nota: el atleta no debe salir del círculo o pasillo hasta tanto el artefacto no halla tocado suelo. El primer contacto con el tope de aro o el suelo fuera del círculo o pasillo se considera dejar (abandonar) el círculo o pasillo.

Tarea	
Número tarea: 6	**Número y nombre historia:** HU 4 archivar datos de la competencia.
Nombre tarea: registro de intentos**.**	
Tipo de tarea : Desarrollo	**Puntos estimados: 1**
Fecha inicio:	**Fecha fin**:
Programador responsable:	
Descripción: Una vez el atleta realice un intento se registrará si es valido o no en la altura correspondiente, si el atleta renuncia a una altura determinada pues no se marca nada, solo se registra la próxima marca y la efectividad del intento. Esto ocurrirá hasta que sea determinado claramente el ganador de la disciplina ya sea por mayor altura alcanzada o por ser el único con todos los intentos validos.	

Los empates se resolverán como sigue:

(a) El atleta con menor número de saltos a la altura en que se produzca el empate será el que obtenga el mejor puesto.
(b) Si el empate aún continúa, el atleta con el menor número total de intentos nulos (fallos) durante la competición hasta la altura últimamente franqueada inclusive, obtendrá el mejor puesto.

Si aún persiste el empate:

(1) Si afecta al primer puesto, los atletas empatados deberán disponer de un salto más en la altura que sea, determinada por el Artículo 181.1, la última saltada por los atletas empatados y si no se lograra una decisión se hará elevar o descender el listón si los atletas franquean o fallan respectivamente, 2 cm. en el salto de altura y 5 cm en el salto con pértiga, debiendo hacer un intento en cada altura hasta que el empate sea resuelto.

Los atletas así empatados tienen que saltar en cada ocasión en que se resuelva el empate. (Ver ejemplo).

(2) Si afecta a cualquier otro puesto, los atletas serán clasificados en el mismo lugar en la competición.

Salto de altura (Ejemplo)

Alturas anunciadas por el Juez Jefe al comienzo de la competición: 1,75 m.;1,80 m.;1,84 m.; 1,88 m.;1,91 m.;1,94m.;1,97 m.;1,99 m...

Atleta	Alturas							Nulos	Saltos de desempate			Puestos
	1,75 m.	1,80 m.	1,84 m.	1,88 m.	1,91 m.	1,94 m.	1,97 m.		1,91 m.	1,89 m.	1,91 m.	
A	O	XO	O	XO	X-	XX		2	X	O	X	2
B	-	XO	-	XO	-	-	XXX	2	X	O	O	1
C	-	O	XO	XO	-	XXX		2	X	X		3
D	-	XO	XO	XO	XXX			3				4

O = Salto válido X = Intento Nulo - = Nulo

Bibliografía

1. Framework Hibernate. [Online] http://www.Hibérnate.org.

2. Base de datos. [Online] http://www.postgresql.org.

3. Eclipse. [Online] http://150.244.56.228/descargas_web/cursos_verano/20040801/Jesus_Montero/documentacion_eclipse.pdf.

4. Framework Hibarnate. [Online] http://www.unife.edu.pe/ing/desarrollo.doc.

5. Framework Ibatis. [Online] http://ibatis.apache.org/

6. Hibernate. [Online] http://www.javahispano.org/contenidos/archivo/77/ManualHibernate.pdf

7. metodologia XP. [Online] http://ones.sourceforge.net/proyecto/ch05.html.

8. Metodologia XP. [Online] http://www.mografias.com.

9. Pgadmin III. [Online] http://www.sourcefoge.not/

10. Pgadmin III. [Online] http://www.pgarmin.org.

11. XP. [Online] http://programacionextrema.tripod.com/index.htm.

12. Visual Paradigm. [Online] http://www.fileheaven.com/descargar/visual-paradigm-for-uml-professional-edition/23018.htm.

13. Visual Paradigm. [Online] http://www.freedowloadmanager.org/es/dowloads/Paradigma Visual para UML (M%3%8D) 14720 p/.

14. Soporte PostgreSQL. [Online] http://www.dbas.com.ar.

15. Rationa rose. [Online] http://www.rationar.com.ar/herramientas/roseenterprise.html.

16. PostgreSQL. [Online] http://www.postgresql.org/.

17. XP. [Online] http://www.xprogramming.com/software.htm.

18. XP. [Online] http://programacionextrema.tripod.com/index.htm.

19. EMS SQL Manager. [Online] http://www.freedownloadmanager.org/es/downloads/SME_Gerente_de_SQL_2005_para_PostgreSQL_36037_p/

Glosario de términos

Feedback: es el continuo seguimiento que recibimos a la hora de desarrollar en un entorno de desarrollo ágil.

Test: son las pruebas de aceptación que se crean antes de realizar el código y por los cuales tiene que pasar y aprobar para poder tenerlo como un código valido, y así formar parte del sistema.

Relases: Entregables que son resultado de la programación del sistema.

Usanza: Usabilidad de procedimientos y métodos ya conocidos.

Multiplataforma: puede ser ejecutado en diferentes Sistemas Operativos (SO).

Toolkit: Es una colección de herramientas integradas que permiten automatizar un conjunto de tareas de algunas de las fases del ciclo de vida del sistema.

Plug-ins: Programas que deben instalarse en los ordenadores clientes para que sus navegadores puedan reconocer y procesar determinados tipos de archivo.

Rollback: Esto ocurre cuando hay algún tipo de fallo en una transacción utilizando Bases de Datos, todos los datos que no hayan sido guardados (dar commit), regresan a su estado original, evitando así la (soporte de PsgreSQl) perdida de datos cuando un cliente falla.

Foto-Finish: mecanismo implementado para determinar la llegada de los atletas a la meta.