



Universidad de las Ciencias Informáticas
Facultad 08

*Sistema para la Gestión de la Información de los Juegos
Deportivos Inter-Facultades Versión 2.0*

Trabajo de Diploma para optar por el Título de
Ingeniero en Ciencias Informáticas

Autores:

Julio Antonio Kelly Naranjo
Julio César Cañabate Gutierrez

Tutor:

Ing. Leonardo San Roman Labaut.

Ciudad de La Habana

Mayo del 2010

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a que haga uso ilimitado con este. Queda confirmada la presente declaración a los ____ días del mes de _____ del 2010.

Autores:

Julio Antonio Kelly Naranjo

Julio César Cañabate Gutierrez

Tutor:

Ing. Leonardo San Roman Labaut

El éxito no se logra sólo con cualidades especiales, es sobre todo un trabajo de constancia, de método y de organización.

J. P. Sergent

AGRADECIMIENTOS

A todos los que lucharon para que hoy exista esta gran obra que es la Revolución Socialista, que nos ha permitido graduarnos hoy como Ingenieros.

A nuestros padres, pues sin ellos no estudiáramos hoy aquí.

A nuestra familia, por alentarnos y darnos su apoyo siempre que lo necesitábamos.

A todos los que brindaron su apoyo en la realización de este trabajo, en especial a Héctor y Daniel.

A nuestro tutor por el apoyo desinteresado en el presente trabajo.

A los que confiaron y a los que no lo hicieron, a todos:

Gracias

Julio Antonio Kelly:

A mis padres, María Victoria y Julio Antonio por ser los mejores padres del mundo, porque cada día que pasa están ahí para ayudarme y aconsejarme pues sin el apoyo de ellos no hubiera podido ser posible nada en absoluto.

A mi abuelita Rebecca que la quiero con mi alma y sé que está esperando este día con todas las ansias de su corazón.

A mis hermanos Juan Carlos y Ronaldo Javier, que sigan estudiando y que se esfuercen cada día más por ser mejores.

Y por último a todos mis compañeros que a lo largo de estos 5 años han estado siempre conmigo, en las buenas y en las malas.

A todos ellos va dedicado este trabajo.

DEDICATORIA

Julio César:

A mis padres Adys Gutierrez Quiñones y Adolfo de Jesús Cañabate Mecías, por su amor, dedicación, por estar siempre a mi lado, por todo lo que han hecho por mí durante más de 23 años.

A mi única hermana, a la incansable, la luchadora.

A mi patria, a los hombres y mujeres que por décadas lucharon para que hoy exista una Revolución Socialista que sin ella no hubiese alcanzado mi mayor sueño: llegar a la Universidad, y hoy ser Ingeniero.

A mi familia, por el aliento, la preocupación, por los momentos felices, por el apoyo cuando los he necesitado.

A mis compañeros de estudio, a los que me ayudaron durante todos estos años.

En la Universidad de las Ciencias Informáticas se han creado varios espacios para el disfrute y esparcimiento de toda su comunidad. Uno de los más representativos lo constituyen los Juegos Deportivos Inter-Facultades, sin embargo, los software que se han utilizado para el respaldo de dicha actividad no gestionan de forma correcta toda la información. Es por esto que a finales del curso 2008-2009, se crea un software con el objetivo de gestionar los procesos relacionados con la información de los Juegos Deportivos Inter-facultades, aunque este sigue presentando un gran problema: para acceder al mismo se necesitan roles administrativos, haciéndolo invisible para la comunidad universitaria. Es por ello, que surge así la necesidad de realizar una aplicación que muestre a la población universitaria la información inherente al acontecer deportivo que se gestiona con la aplicación creada en el curso 2008-2009. Para lograrlo se analiza el flujo actual de los procesos presentes en la conformación de los Juegos Inter-Facultades. También se realizó un estudio del software “Sistema para la Gestión de la Información de los Juegos Deportivos Inter-facultades”. A partir de los análisis anteriores se identifica la información que se desea mostrar para finalmente conformar la propuesta de solución que garantice obtener el resultado esperado. Para dicha propuesta se utilizaron las denominadas tecnologías Web, conjuntamente con una metodología de desarrollo de software, en este caso eXtreme Programming, las cuales permitieron la elaboración de un subsistema informático, cuyo título es: “Sistema para la Gestión de la Información de los Juegos Deportivos Inter-Facultades Versión 2.0”.

Índice

Introducción:.....	1
Capítulo I Fundamentación Teórica.....	6
1.1 Introducción.....	6
1.2 Fundamentación del tema	6
1.2.1 Conceptos fundamentales.....	6
1.2.2 Sistemas informáticos similares.....	7
1.3 Tecnologías actuales	8
1.3.1 Aplicaciones Web	8
1.3.2 Lenguajes de programación Web	9
1.3.3 Sistemas Gestores de Base de Datos	13
1.3.4 Servidores Web	15
1.3.5 Metodologías de desarrollo de software	16
1.3.6 Framework	21
1.3.7 Arquitectura.....	24
1.3.8 Herramientas web utilizadas en la propuesta de solución	26
1.4 Conclusiones.....	28
Capítulo II Características del Sistema.	29
Elaboración y planificación.....	29
2.1. Introducción.....	29
2.2. Flujo actual de los procesos.....	29
2.3. Objeto de automatización	29
2.4. Propuesta del sistema.....	30
2.4.1. Personal relacionado con el sistema.....	31
2.5. Fase de Exploración.....	31
2.5.1. Historias de Usuario (HU).....	31
2.6. Fase de Planificación	34
2.6.1. Estimación de esfuerzo por Historias de Usuario	35
2.6.2. Plan de iteraciones.....	35
2.6.3. Plan de duración de las iteraciones.....	36
2.6.4. Plan de entrega	37
2.7 Conclusiones.....	38
Capítulo III: Implementación y Prueba.....	39
3.1. Introducción.....	39
3.2. Diseño del sistema	39
3.2.1. Tarjetas CRC	40
3.2.2. Modelo de datos	47
3.3. Implementación	48
3.3.1. Iteración 1.....	48
3.3.2. Iteración 2.....	53
3.4. Prueba	56
3.4.1. Pruebas de aceptación	57
3.5. Conclusiones.....	62
Conclusiones.....	63
Recomendaciones	63
Referencias Bibliográficas	64
Bibliografía.....	65

Introducción

En Cuba, luego del triunfo revolucionario en 1959, urge la necesidad de formar profesionales en todos los ámbitos y materias, comenzando por una Campaña de Alfabetización en donde utilizó todo aquel que tuviera un sexto grado o al menos supiera leer y escribir, para erradicar de una vez el analfabetismo existente y predominante en toda la Isla.

Luego de esa gran campaña y una vez logrado el objetivo, comenzaron a desarrollarse facultades obreros-campesinas, tecnológicos, institutos de diversas materias, etc., con el objetivo de aumentar el nivel profesional en el país, muy escaso en aquel entonces, debido al robo o compra de cerebros del que se era víctima.

Muchos jóvenes ingresaron a las universidades existentes en aquellos momentos de la historia, universidades que se fueron nutriendo y perfeccionando con los mismos egresados y mejores expedientes.

Hoy, la historia continúa, o mejor, esa revolución educacional sigue sin detenerse, con más ímpetu y fuerza, evidenciándose con los nuevos planes de la Revolución Socialista, a lo que nuestro Comandante en Jefe llamase “Batalla de Ideas”.

En ese calor educacional, en donde hoy todo joven cubano puede aspirar a una carrera universitaria, nace la Universidad de las Ciencias Informáticas, un centro de altos estudios considerado como de “Nuevo Tipo”, lugar en el que la docencia y la práctica van de la mano, en donde no sólo se forma a un profesional en el ámbito científico y tecnológico, sino que la integralidad de ese individuo va más allá, nutrida por la cultura, la investigación y el deporte. Magníficos ejemplos de ello se puede evidenciar con los Festivales de Artistas Aficionados, Eventos Investigativos, Juegos Deportivos, entre otros, manteniendo estos últimos en actividad al 100 % de la población universitaria, mayormente si se desarrollan entre Facultades, en donde la rivalidad por defender el color de cada una de ellas se convierte en toda una fiesta.

Los Juegos Deportivos Inter-Facultades, edición tras edición, son perfeccionados y enriquecidos por otros deportes que se les van incorporando, al igual que por nuevos deportistas que engrosan las filas de los que, si de músculo, agilidad, destreza e inteligencia se trata, lo han dado todo, quienes han hecho historia aquí y en otras Universidades del País, participando en las Olimpiadas del Deporte Cubano.

En nuestra Universidad están creadas las condiciones para que los Juegos Deportivos se realicen con la mayor calidad, dependiendo en gran medida de los medios tecnológicos que se disponen, sirviendo estos últimos de apoyo para la divulgación de la información.

En el curso 2006-2007 se realizó un software llamado Sistema de Gestión Deportiva para la Universidad de las Ciencias Informáticas, utilizándose en ese período para la gestión de la información de los Juegos Inter-Facultades, pero al siguiente año este software se dejó de utilizar debido a que incumplía con políticas y estándares establecidos por la Universidad, que estipulan que las aplicaciones realizadas con herramientas propietarias no se deben de publicar, ya que a nuestro país se le imposibilita el acceso legal de estas herramientas, debido a restricciones establecidas por compañías propietarias, fortalecidas por el embargo financiero que establece el gobierno de los EEUU. Por lo antes expuesto se le da gran importancia al uso exclusivo del software libre para desarrollar aplicaciones de uso dentro y fuera de la Universidad.

En el curso 2007-2008 se utilizó una aplicación que se realizó días antes de los juegos, la cual era totalmente estática y no ofrecía todos los servicios y prestaciones que demandaba nuestra comunidad, además, era muy difícil de gestionar toda la información que se mostraba en la misma, debido a que fue desarrollada por personas con pocos conocimientos informáticos.

En el curso 2008-2009 se empleó para los Juegos Inter-Facultades un portal montado en CMS Joomla, teniendo como principal problema que no gestionaba de forma dinámica toda la información que se debía manejar en unos juegos deportivos y dicha información debía ser actualizada de forma manual por los operarios, lo cual provocaba que se cometieran errores y mucha pérdida de tiempo. También hay que destacar que solo se utilizaban los beneficios del CMS para la parte administrativa y del diseño de la aplicación. Al finalizar la edición gran parte de esa información desapareció o al menos no está disponible para todo aquel interesado del acontecer pasado.

Al concluir el curso, se terminó y presentó una aplicación para la gestión de los juegos, cuyo alcance no tiene un beneficio para toda la población universitaria, realizada solo con fines administrativos, en donde el/los administrador/es gestionaría/n la información relacionada con los Juegos Deportivos Inter-Facultades. A pesar de las funcionalidades anteriormente dichas, brindadas por el software anterior, siguen existiendo problemas como:

- No existe un medio donde se pueda consultar la información de las diferentes ediciones de los Juegos Deportivos.
- No se pueden realizar análisis cualitativos y cuantitativos sobre las condiciones deportivas de los estudiantes en cada nueva edición.
- No se pueden realizar análisis cualitativos y cuantitativos sobre las condiciones deportivas de las facultades a asumir en nuevos cursos.

- No se pueden utilizar datos como estadísticas para un determinado deporte.

Para resolver los problemas anteriores se propone una nueva versión de un subsistema informático que gestione la información deportiva de los Juegos Inter-Facultades en la Universidad de las Ciencias Informáticas.

A partir de la situación anteriormente expuesta, unida a la necesidad vigente en la UCI de crear plataformas libres como parte de la estrategia nacional de migración hacia este tipo de tecnologías, se propone el siguiente **problema a resolver**:

➤ ¿Cómo facilitar con rapidez y organización la gestión de la información inherente a los Juegos Deportivos Inter-Facultades de la Universidad de las Ciencias Informáticas?

Objetivo General:

Desarrollar una segunda versión del Subsistema para la Gestión de la Información de los Juegos Deportivos Inter-Facultades, que permita gestionar y mostrar la información deportiva en la Universidad.

Objeto de Estudio:

Sistemas para gestionar la información de Juegos Deportivos.

Objetivos específicos:

- 1- Estudiar toda la información inherente al acontecer de los Juegos Deportivos Inter-Facultades en la Universidad de Ciencias Informáticas.
- 2- Realizar la exploración y la planificación de un subsistema que gestione la información de los Juegos Deportivos Inter-Facultades en la UCI.
- 3- Implementar un subsistema que gestione la información de los Juegos Deportivos Inter-Facultades en la UCI.

Campo de Acción:

Subsistema para gestionar la información de los Juegos Inter-Facultades en la Universidad de las Ciencias Informáticas.

Resultados Esperados:

Si se desarrolla un subsistema que gestione toda la información deportiva en la Universidad de Ciencias Informáticas, se lograría mayor rapidez y organización en el control, procesamiento y actualización de la información, además de que se eliminaría la desinformación en la comunidad sobre el acontecer de los anteriores juegos deportivos inter-facultades.

Método Científico:

En la investigación se hizo un análisis histórico y lógico de los sistemas informáticos vinculados con el deporte que existen en el país. Utilizando el método de la observación se determinó si los existentes en la Universidad gestionan de forma correcta la información relacionada con los Juegos Deportivos Inter-Facultades. También se utilizó la técnica de la entrevista para obtener información de las necesidades de la Universidad relacionadas con la gestión de la información de dichos juegos.

Tareas a Desarrollar:

Para el desarrollo de la aplicación se han trazado las siguientes tareas:

- ✓ Leer, analizar y procesar la información relacionada con la Gestión de la información de los Juegos Deportivos Inter-facultades.
- ✓ Realizar la planificación para la posterior implementación del subsistema.
- ✓ Implementar el subsistema “Sistema para la Gestión de la Información de los Juegos Deportivos Inter-Facultades Versión 2.0”.
- ✓ Estructurar la investigación realizada en el formato requerido para obtener una buena documentación.

Estructura capitular:

Este trabajo cuenta con 3 capítulos:

En el capítulo 1, se hace referencia a la fundamentación teórica, donde se realiza un estudio de la actualidad y necesidad del presente trabajo, así como de las tecnologías, herramientas y lenguajes de programación a emplear durante el desarrollo de la aplicación. Se fundamenta, además, la metodología de desarrollo de software utilizada para guiar la investigación.

En el capítulo 2, se explican los procesos vinculados al campo de acción, momento a partir del cual se comienza a hacer el análisis del sistema a desarrollar. Atendiendo a las características que se identifiquen como necesarias para el desarrollo de dicho sistema, se procede a asignar responsabilidades a los futuros usuarios del mismo atendiendo a sus competencias. Posteriormente se detallan los artefactos que se generan en la fase de Exploración y Planificación, centrandose la atención en las historias de usuario. Se construyen otros artefactos como el Plan de Iteración y el Plan de Entregas que son de suma importancia para el desarrollo de la aplicación.

En el capítulo 3, se exponen los aspectos relacionados con el diseño del sistema. Se describen las tareas propuestas para cada historia de usuario relacionadas a la fase de implementación y las pruebas de aceptación realizadas a solicitud del cliente.

Capítulo 1 Fundamentación Teórica

1.1 Introducción

En el mundo de la informática, específicamente en la producción de software, es común observar como lo que hasta ayer parecía la mejor solución hoy resulta obsoleto o atrasado, y mañana surge otra mejor. Esta idea lejos de agobiar a los desarrolladores les incita a adentrarse, cada vez más, dentro del excitante entorno que rodea a la mundialmente denominada "Industria del Software".

En el presente capítulo, se exponen conceptos esenciales asociados a las diversas y novedosas tecnologías de desarrollo existentes para dar solución al problema planteado.

1.2 Fundamentación del tema

1.2.1 Conceptos fundamentales

Gestión de la información

El factor esencial del progreso es el conocimiento, dependiente en gran medida de la información. Uno de los principales problemas es su exceso, debido, entre otras causas al surgimiento y desarrollo de Internet. Es frecuente encontrar un número significativo de publicaciones redundantes y de baja calidad mezcladas con otras importantes y sólidas, difíciles de hallar entre la información ruidosa que la "envuelve". La información es un elemento fundamental para el desarrollo. La gestión de la información se ocupa de los resultados finales, no sólo de citas y localizaciones. (1)

Finalmente, la **gestión de la información** posibilita los mecanismos de búsqueda de información de forma eficiente y en menor tiempo posible.

Deporte

El deporte es entendido como actividad física en forma de juego, basado en la determinación de quien lo hace mejor y más rápido. También es utilizado como medio de educación y formación de los jóvenes o como una preparación para las artes guerreras. En nuestro país, el deporte es derecho de todos, practicándose desde edades tempranas en los centros escolares y deportivos, propiciando el desarrollo de capacidades y habilidades motrices en los ciudadanos.

Juegos Deportivos Inter-Facultades

Los Juegos Deportivos Inter-Facultades o “Juegos Mellas”, como se le determinó en el 2009, son toda una fiesta en la UCI, en ellos participan estudiantes de todas las Facultades de la Universidad, participación que incrementó con la representación de sus Facultades Regionales en Artemisa, Ciego de Ávila y Granma, incorporándose estas en el 2008. Es una de las actividades más esperada del año, que promueve la práctica de ejercicios físicos y la formación de valores en función de mejorar la calidad de vida de los estudiantes y el desarrollo integral de su personalidad. Al finalizar, se captan los deportistas más destacados para que represente a la Universidad en los eventos deportivos en que se realizan en distintos lugares del país.

1.2.2 Sistemas informáticos similares

El deporte se ha convertido no solo en una actividad de entretenimiento, cultura física y competencia, sino que también, ha pasado a jugar un papel determinante en la esfera económica y lucrativa sobre todo en los países del primer mundo. La informática progresivamente se ha ido incorporando a diversos ámbitos, y el mundo del deporte no es ajeno a este fenómeno, presentándose como un campo en el que esta ciencia tiene enormes posibilidades de aplicación. Ante esta realidad, la inserción de sistemas informáticos en el ámbito deportivo es un hecho comprobado.

En el plano internacional, son muchos los sistemas informáticos especializados en la gestión de la información relacionada a eventos deportivos entre los que se pueden mencionar el portal de la ESPN - <http://espn.go.com/> - el cual brinda cobertura a numerosas ligas o eventos deportivos que tienen lugar en el mundo. Se caracteriza por poseer una navegación simple y ligera lo cual sin dudas beneficia a los usuarios. Brinda, además, la posibilidad de filtrado de datos e informaciones atendiendo a muy variados criterios. Existen otros portales, similares al anterior, como los pertenecientes a consorcios de la información como Yahoo - <http://espanol.sports.yahoo.com/> - y Fox - <http://msn.foxsports.com/> -.

En Cuba, los sistemas informáticos relacionados a la gestión de información deportiva que existen, no se igualan al nivel internacional. Se destaca el sistema **Estudio de Contrarios de Béisbol**, el cual facilita a los entrenadores el trabajo estadístico y por ello es de vital importancia para esta disciplina deportiva. Por otra parte, en la UCI, se cuenta con un portal muy popular conocido como **beisbol.uci.cu** el cual da seguimiento a la Serie Nacional de Béisbol brindando disímiles funcionalidades.

Las aplicaciones nacionales antes mencionadas se caracterizan por brindar una solución eficiente a los problemas por los cuales fueron concebidas, sin embargo, como se aprecia, restringen su dominio a gestionar la información referente a un solo deporte.

1.3 Tecnologías actuales

1.3.1 Aplicaciones Web

Se denomina aplicación Web a aquellas aplicaciones a las cuales acceden los usuarios a través de Internet o Intranet mediante un navegador. (2)

En lo que respecta a su arquitectura, se distinguen dos lados, uno es el cliente donde se encuentra el usuario final interactuando, por intermedio de un navegador, con la aplicación localizada al otro lado o servidor en donde residen realmente los datos, reglas y lógica de la aplicación. Precisamente dicha arquitectura potencia la popularidad de las aplicaciones Web debido a lo práctico que resulta el navegador Web como cliente ligero así como a la facilidad para gestionarlas sin distribuirlas e instalarlas a grupos de usuarios potenciales. Si bien es cierto que la arquitectura cliente-servidor ofrece múltiples ventajas es una realidad que carece de la riqueza gráfica presente en las aplicaciones de escritorio que dan mayor fluidez al trabajo de los usuarios, sin embargo, este problema se ha ido resolviendo con la incorporación de tecnologías como AJAX. De este modo las aplicaciones Web buscan alternativas que le permitan ofrecer todas sus ventajas pero con la posibilidad de ofrecer controles visuales más amigables al trabajo del usuario.

Con la división del problema en dos partes, se logra centralizar la administración en general a un solo lado: el servidor, resolviendo una gran cantidad de problemas anteriormente encontrados en las aplicaciones de escritorio mono usuario, como son:

- Duplicidad de datos por la falta de unificación de los mismos.
- Diseminación de la información y lógica en muchas partes (cada computador que la use).
- Falta de portabilidad de la aplicación a diferentes sistemas operativos.
- Traumas a la hora de realizar actualizaciones o correcciones al programa ya que las instalaciones están diseminadas.
- La administración de la seguridad, controlando el acceso a los usuarios a información no relevante o privada, es un caos.
- Dificultad para configurar cada una de las instalaciones dependiendo de las necesidades de cada usuario. (3)

1.3.2 Lenguajes de programación Web

Para utilizar un lenguaje Web, se debe conocer con exactitud qué es lo que se quiere hacer y si el lenguaje en cuestión lo permite o no. Los lenguajes de programación para la Web se dividen en dos grupos a tono con la propia arquitectura cliente-servidor: los lenguajes del lado del Cliente y los lenguajes del lado del Servidor.

Los lenguajes del lado del cliente, entre los que se encuentran XHTML, Javascript y CSS, son independientes del servidor lo cual significa que pueden ser “digeridos” directamente por el servidor y no necesitan pre-procesamiento. Entre los lenguajes del lado del servidor, los cuales se caracterizan por desarrollar la lógica de negocio dentro del servidor además de encargarse del acceso a bases de datos y al tratamiento de la información, se distinguen Python, Perl y PHP.

Lenguajes del Lado del Cliente

XHTML significa Lenguaje extensible de marcación de hipertexto (Extensible Hypertext Markup Language). Este lenguaje ofrece la norma clásica para crear páginas web, el Lenguaje de marcación de Hipertexto (HTML, HyperText Markup Language), y la nueva norma para datos descriptivos, el Lenguaje de marcación extensible (XML, Extensible Markup Language). (4)

Precisamente, XHTML, surge basado en la necesidad de reemplazar al HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos. XHTML, al estar orientado al uso de un etiquetado correcto, exige una serie de requisitos básicos a cumplir en lo que a código se refiere. Entre estos requisitos básicos se puede mencionar una estructuración coherente dentro del documento donde se incluirían elementos correctamente anidados, etiquetas en minúsculas, elementos cerrados correctamente, atributos de valores entrecomillados.

Javascript

Se trata del lenguaje del lado del cliente más utilizado gracias a su compatibilidad con la mayoría de los navegadores. Mediante su uso se pueden crear efectos especiales en las páginas y definir interactividades con los usuarios. Gran parte de la programación en este lenguaje se dirige a escribir funciones que responden a determinados eventos como el movimiento del mouse, la utilización de teclas, la carga de páginas, entre otros. También es muy utilizado a la hora de validar los datos introducidos por los usuarios a través de los formularios. Es necesario precisar que existen dos tipos de Javascript: por un lado está el que se ejecuta en el cliente, este es el Javascript propiamente dicho, aunque técnicamente se denomina Navigator Javascript y por el otro el que se ejecuta en el servidor,

más reciente y denominado LiveWire Javascript.

CSS (Cascading Style Sheets)

CSS es un lenguaje del lado del cliente creado para describir cómo se mostrará un documento, definido con HTML o XHTML, en pantalla o cómo se va a imprimir e incluso cómo será pronunciada la información presente en el documento a través de un dispositivo de lectura. Mientras que el lenguaje XHTML se utiliza para marcar los contenidos, es decir, para designar lo que es un párrafo, lo que es un titular o lo que es una lista de elementos, el lenguaje CSS se utiliza para definir el aspecto de todos los contenidos, es decir, el color, tamaño y tipo de letra de los párrafos de texto, la separación entre titulares y párrafos, la tabulación con la que se muestran los elementos de una lista, etc.(5) CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

Lenguajes del Lado del Servidor

Python

Es un lenguaje de programación, manejado como proyecto de software libre, de propósito general, o sea, permite la creación de todo tipo de aplicaciones incluyendo los sitios web. Habitualmente se le compara con Perl y sus usuarios consideran que es más limpio para programar, aunque esto no es más que un punto de vista de los mismos. Es un lenguaje multiplataforma y multiparadigma, esto último permite a los programadores adoptar un estilo de programación particular. Python es visto como un lenguaje exitoso debido a su facilidad de aprendizaje, su orientación a programadores promedio y la limpieza de su código.

Perl

Perl es un acrónimo de Practical Extracting and Reporting Language, lo cual se traduce en que se trata de un lenguaje práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros. Es un lenguaje libre asociado a la plataforma Unix lo cual no indica que no esté disponible en otros Sistemas Operativos como Windows. (6) Al ser Perl un lenguaje de programación interpretado el código contenido en sus scripts no se compila sino que cada vez que se quiere ejecutar se lee el código y se pone en marcha interpretando lo que hay escrito.

PHP

Es un lenguaje de programación interpretado usado generalmente para la creación de contenido para sitios o aplicaciones Web. Sus siglas representan un acrónimo recursivo que significa “Hypertext Preprocessor” aunque vale destacar que inicialmente se denominó Personal Home Page. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl a lo cual se incorporan características específicas.

PHP es un producto de software libre, debido a esto cuenta con la colaboración de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparen rápidamente. Debido al intenso trabajo de la extensa comunidad de desarrolladores que posee, PHP se actualiza continuamente con mejoras y extensiones de lenguaje para ampliar sus capacidades, manteniendo de igual manera una amplia, práctica y actualizada documentación (7).

PHP ha sido diseñado de forma muy modular y el surgimiento de nuevas librerías contribuye a la notable sencillez que posee. Toda esta funcionalidad está basada en librerías que en su mayor parte no han sido desarrolladas por el equipo de PHP (8).

Entre sus características fundamentales están:

- Libre y Gratuito: Al tratarse de software libre puede descargarse y utilizarse en cualquier aplicación, personal o profesional, de manera completamente libre.
- Rapidez de ejecución.
- Mantiene un bajo consumo de recursos de máquina.
- Gran seguridad: Muy poca probabilidad de corromper los datos.
- Permite las técnicas de Programación Orientada a Objetos.
- No requiere definición de tipos de variables.
- Gran popularidad: Existe una gran comunidad de programadores y desarrolladores que continuamente implementan mejoras en su código, y que se ayudan entre sí cuando se enfrentan con un problema. Posee una amplia documentación en Internet, incluyendo una gran variedad de ejemplos y de ayudas.
- Eficiencia: Con escaso mantenimiento puede soportar sin problemas millones de visitas diarias.

- Sencilla integración con múltiples bases de datos: Esencial para una Página Web verdaderamente dinámica es una correcta integración con bases de datos, PHP trabaja con bases de datos muy conocidas como MySQL, PostgreSQL, Oracle, dbm, filePro, interbasem o cualquier otra de datos compatible con ODBC(Open Database Connectivity Standard).
- Versatilidad: PHP puede utilizarse con la mayoría de sistemas operativos como Unix (Linux, Solares, FreeBSD, y otros), como con Windows.
- Gran número de funciones predefinidas: A diferencia de otros lenguajes de programación, PHP fue diseñado para el desarrollo de páginas web dinámicas, por ello está dotado de un gran número de funciones que posibilitan la simplificación de códigos complejos o poco complejos que se reducen a una palabra.

La versión 5 de PHP provee a los desarrolladores de un magnífico trabajo con el Paradigma Orientado a Objetos que permite la reutilización de código entre otras facilidades, de este modo se coloca más a tono con las exigencias de la programación moderna.

Selección de los Lenguajes de Programación Web

Luego de realizado el análisis de varios de los lenguajes de programación utilizados en la actualidad para el desarrollo de aplicaciones Web y atendiendo a que los mismos están divididos en Lenguajes del lado del Cliente y Lenguajes del lado del Servidor se resume:

Del lado del Cliente

La combinación de los lenguajes XHTML, Javascript y CSS es la más utilizada en la actualidad para desarrollar el contenido referente a la parte del cliente en las aplicaciones Web debido a que los tres se complementan de forma ideal brindando una amplia gama de posibilidades a los desarrolladores.

XHTML sin dudas es una versión más acabada del HTML motivo por el cual es ventajoso utilizarlo como lenguaje principal con el objetivo de describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

Javascript sigue siendo uno de los lenguajes imprescindibles a la hora de crear interacciones con el usuario mediante el manejo de eventos. Actualmente con el surgimiento del AJAX, las prestaciones que brinda se han elevado considerablemente.

Por otra parte, si se quiere controlar el estilo y formato de múltiples páginas web de manera simultánea, separando el estilo de la presentación de los documentos XHTML, sin dudas el uso de las Hojas de Estilo en Cascada (CSS) se hace necesario.

Por tanto, se seleccionan los lenguajes **XHTML**, **Javascript** y **CSS** como Lenguajes de Programación del lado del Cliente atendiendo, además, a la experiencia que se tiene sobre su uso.

Del lado del Servidor

Como se ha podido analizar, Perl, Python y PHP, son excelentes lenguajes de programación, pero entre ellos, PHP se adecua mejor a las exigencias de la investigación, además, a diferencia de los otros dos lenguajes mencionados, fue diseñado desde cero con el objetivo de desarrollar aplicaciones Web. Las tareas más habituales para la realización de estos tipos de aplicaciones pueden hacerse con PHP de forma fácil, rápida y efectiva. La curva de aprendizaje de PHP no es para nada elevada por cuanto los resultados de su uso son rápidamente observables. PHP no obliga al uso de determinada metodología al programar, como muchos otros lenguajes tampoco, sin embargo, los desarrolladores son libres de elegir para su trabajo cualquier técnica que les permita mantener su código ordenado.

Por tanto, se selecciona el lenguaje **PHP** como Lenguaje de Programación del lado del Servidor.

1.3.3 Sistemas Gestores de Base de Datos

Los Sistemas Gestores de Bases de Datos (SGBD) son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. Aun cuando en la actualidad existe una gran variedad de SGBD, las principales funciones que deben cumplir se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, así como evitar redundancias e inconsistencias y mantener la integridad.

A continuación se realiza un análisis, desde la perspectiva del software libre, de los SGBD MySQL y PostgreSQL como las herramientas para administrar y desarrollar sistemas de información más comunes dentro del desarrollo Web.

MySQL

Se trata de un sistema de gestión de base de datos relacional, multihilo y multiusuario. Constituye el servidor de bases de datos relacionales más popular y es desarrollado por la empresa MySQL AB, la cual desde enero del 2008 pertenece a Sun Microsystems. MySQL es desarrollado como software libre siguiendo un esquema de licenciamiento dual, o sea, se ofrece bajo la GNU GPL para cualquier uso

compatible con esta licencia. Sin embargo, si no se está de acuerdo con la misma o se necesita incorporar MySQL en una aplicación comercial es posible comprar una licencia específica que permita este uso.

Contrario a otros proyectos de software libre, como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es propiedad y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.(9)

PostgreSQL

Está considerado el SGBD de software libre más avanzado del mundo, publicado bajo la licencia BSD. PostgreSQL proporciona un gran número de características que normalmente sólo se encontraban en los SGBD comerciales de alto calibre tales como Oracle. Es un SGBD objeto-relacional, ya que aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Su avanzada funcionalidad se pone de manifiesto con las consultas SQL declarativas, el control de concurrencia multiversión, soporte multiusuario, transacciones, optimización de consultas, herencia y valores no atómicos (atributos basados en vectores y conjuntos). Sirve de soporte a los lenguajes más populares como PHP, C, C++, Java, Python, Ruby, entre otros, además de soportar un número ilimitado de bases de datos.

Entre sus características fundamentales respecto a otros gestores están:

- Gran escalabilidad. Es ajustable al número de procesadores y a la cantidad de memoria que posee el sistema de forma eficiente, por este motivo es capaz de soportar una mayor cantidad de peticiones simultáneas. Teniendo en cuenta esto, es vital en la universidad, ya que no se requiere de un avanzado sistema de cómputo para trabajar con él.
- Tiene la capacidad de almacenar procedimientos almacenados en la propia base de datos.
- Multiusuario, con arquitectura cliente-servidor y control de privilegios de acceso.
- Los tipos internos han sido mejorados, incluyendo nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales.

Selección del Sistema de Gestión de Base de Datos (SGBD)

En resumen, tanto MySQL como PostgreSQL cuentan con un magnífico soporte para la Web. El uso de MySQL se ve favorecido cuando el factor velocidad es primordial, mientras que PostgreSQL inclina la

balanza a su favor cuando se busca la estabilidad, integridad y seguridad del sistema, cualidades estas últimas deseadas para dar solución al problema de la investigación.

Por tanto, se selecciona **PostgreSQL** como SGBD atendiendo a sus características.

1.3.4 Servidores Web

Un servidor Web no es más que un programa que proporciona datos en forma de páginas web, hipertextos o páginas XHTML mediante la implementación del protocolo HTTP (Hypertext Transfer Protocol).

De este modo, un servidor Web se mantiene a la espera de peticiones HTTP, que son ejecutadas por un cliente HTTP; lo que se conoce como un navegador Web. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita.

HTTP Apache

Es un servidor de páginas web, de software libre, que posibilita el acceso a las páginas web alojadas en un ordenador. Su diseño le permite ser un servidor Web potente y flexible, funcional en una amplia gama de plataformas y entornos. Precisamente las mencionadas cualidades le reservan un lugar cimero en las preferencias de los desarrolladores, convirtiéndose en uno de los servidores Web más utilizados en el mundo. El servidor HTTP Apache se ubica, hoy en día, por encima de sus competidores, tanto libres como comerciales. El diseño modular que lo caracteriza les permite a los administradores de sitios web elegir que funcionalidades serán incluidas en el servidor seleccionando que módulos se van a usar. Los módulos de Apache se encuentran clasificados en tres categorías:

- **Base:** Módulo con las funciones básicas del Apache.
- **Módulos Multiproceso:** Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.
- **Módulos Adicionales:** Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales están definidas dentro del módulo base mientras que el módulo multiproceso se hace necesario para manejar las peticiones. Se han diseñado varios módulos multiprocesos para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código.

Las funcionalidades restantes del servidor se satisfacen por medio de módulos adicionales. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software.

Este servidor tiene capacidad para servir tanto páginas estáticas como dinámicas a través de otras herramientas soportadas que facilitan la actualización de los contenidos usando bases de datos, ficheros u otras fuentes de información.

Comparte entre muchas de sus características su gratuidad, popularidad, sencillez de manejo y su versatilidad. Es además, utilizado por más del 60% de los sitios del mundo (10).

ZOPE

Es un servidor de aplicaciones Web, de software libre, desarrollado fundamentalmente en Python. Entre las características que lo distinguen se encuentra su base de datos orientada a objetos denominada ZODB (Zope Object Database), la cual almacena objetos ordenados en un sistema similar a un sistema de ficheros, pero cada objeto tiene propiedades, métodos u otros objetos. Esta aproximación difiere bastante de las bases de datos relacionales actuales. No obstante, Zope posee la ventaja de contar con múltiples conectores para las diferentes bases de datos relacionales ofreciendo sistemas básicos de conexión y consulta abstrayéndolos como objetos. (11)

Selección del Servidor Web

Es necesario destacar que tanto Zope como HTTP Apache constituyen en la actualidad excelentes opciones para los desarrolladores Web. Evidentemente, la cualidad más significativa de Zope es su diseño orientado a objetos. HTTP Apache se ve favorecido por su estructuración en módulos, su gran compatibilidad con el lenguaje de programación PHP y la experiencia acumulada sobre su uso tanto a nivel mundial como en la Universidad de las Ciencias Informáticas.

Por tanto, se selecciona **HTTP Apache** como servidor Web.

1.3.5 Metodologías de desarrollo de software

El desarrollo de la denominada “Industria del Software” no conoce límites, la búsqueda constante de nuevas y mejores soluciones así lo demuestran. Las exigencias impuestas a los sistemas de software cada día se hacen mayores, requiriéndose mayor calidad y productividad en menos tiempo por parte de los clientes, lo cual trae como consecuencia la aparición de una serie de problemas. Precisamente para dar solución a dichos problemas surgen las metodologías de software.

Una metodología es el conjunto de técnicas y procedimientos que permiten conocer los elementos necesarios para definir un proyecto de software. Esto significa que al término del mismo se habrá producido lo esperado, en el tiempo esperado y con el costo esperado. Todo desarrollo de software es riesgoso y difícil de controlar, pero si no lleva una metodología de por medio, lo que se obtiene es: clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos.

Es necesario subrayar que en la actualidad existe un gran número de procesos de desarrollo, clasificados en dos grandes grupos atendiendo a sus características: las metodologías robustas y las metodologías ágiles.

A continuación se realiza un análisis de dos de las más famosas y conocidas metodologías. La primera, RUP (Rational Unified Process), se encuentra dentro del grupo de las metodologías pesadas mientras que la segunda, XP (eXtreme Programming), es exponente de las denominadas metodologías ágiles.

RUP (Rational Unified Process)

RUP es uno de los procesos más generales que existe, su finalidad no está restringida a guiar desarrollo de software, sino cualquier tipo de proyecto. Apuesta por alcanzar los objetivos mediante el orden y la documentación, características que lo convierten en el más fiel exponente de las metodologías robustas o pesadas. Las cuatro fases definidas por RUP son:

- Inicio (se define el alcance, costo y el tiempo con el cliente)
- Elaboración (análisis y diseño)
- Construcción (implementación)
- Transición (fin del proyecto y despliegue)

El ciclo de vida de RUP se caracteriza por:

1. **Dirigido por casos de uso:** Los casos de usos reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo). (12)
2. **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo con la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son

necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. (13)

3. **Iterativo e Incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. (14)

En RUP cada una de las actividades se han agrupado en 9 flujos de trabajo, los 6 primeros son comúnmente conocidos como flujos de ingeniería mientras que los tres restantes se denominan flujos de apoyo.

Flujos de Ingeniería:

- **Modelo de Negocio:** Entendiendo las necesidades del negocio.
- **Requerimientos:** Definir el ámbito del sistema, definir una interfaz de usuarios para el sistema enfocada a las necesidades y metas del usuario, establecer y mantener un acuerdo entre clientes y otros involucrados sobre lo que el sistema debería hacer, proveer a los desarrolladores un mejor entendimiento de los requerimientos del sistema, proveer una base para estimar recursos y tiempo de desarrollo del sistema, y brindar una base para la planeación de los contenidos técnicos de las iteraciones.
- **Análisis y Diseño:** Se traducen los requisitos a una especificación que describe cómo implementar el sistema. El análisis consiste en obtener una visión del sistema que se preocupa de ver QUÉ hace, de modo que sólo se interesa por los requisitos funcionales. Por otro lado, el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva CÓMO cumple el sistema sus objetivos.
- **Implementación:** Definir la organización del sistema en términos de Subsistemas de Implementación organizados en capas, implementar los elementos de diseño en términos de “Elementos de Implementación” (ficheros Fuentes, binarios, ejecutables y otros), probar los componentes desarrollados independientemente como unidades, e integrar los resultados producidos por desarrolladores independientes o equipos en un sistema ejecutable.
- **Pruebas:** Verificar que se satisfacen los requerimientos tanto funcionales como no funcionales.

Flujos de apoyo:

- Configuración y administración de cambio: Forma de controlar los artefactos propuestos durante los flujos de trabajo que se llevan a cabo.
- Gestión de proyecto: Gestiona el cumplimiento de objetivos, maneja riesgos y conduce la producción del software hacia la satisfacción de clientes y usuarios.
- Ambiente: Prepara el entorno y herramientas de despliegue que harán posible llevar a cabo el proyecto, además, provee el soporte al equipo de desarrollo durante todo el ciclo de vida del proyecto.
- Despliegue: Produce con éxito distribuciones del producto y lo hace llegar a los usuarios finales.

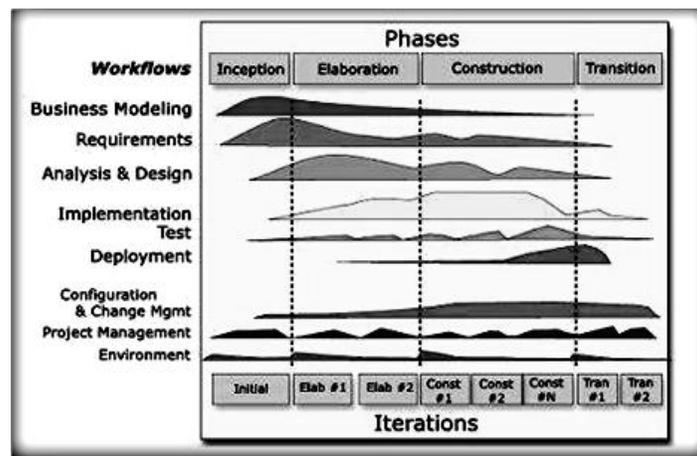


Figura 1: Metodología RUP

XP (eXtreme Programming)

Actualmente con el creciente desarrollo tecnológico y la aparición de nuevos modelos de producción, el uso de las metodologías ágiles gana seguidores. Entre ellas destaca XP, clasificada de metodología ligera por cuanto trata de reducir la complejidad de software orientando el trabajo directamente al objetivo, centrándose tanto en potenciar las relaciones interpersonales de los desarrolladores como en mantener un ambiente de desarrollo colaborativo y agradable, donde la superación continua de los integrantes del equipo se convierta en una práctica cotidiana. Entre sus tendencias resalta la presencia casi a tiempo completo del cliente dentro del proceso de desarrollo, garantizando que sus opiniones contribuyan a una constante retroalimentación desarrolladores-clientes. Quizás dos de los más excitantes valores que posee son: la búsqueda de simplicidad en las soluciones y el coraje requerido para enfrentar los posibles cambios.

XP se basa en UserStories (historias de usuarios), estas historias las escribe el cliente o su representante dentro del equipo y describen los escenarios clave del funcionamiento del software. A partir de estas se planifican las entregas entre el equipo y el cliente. Las entregas son frecuentes, lo que permite mejorar el diseño cada vez que se le añade una nueva funcionalidad. Estas entregas a su vez permiten definir las iteraciones necesarias para cumplir con los objetivos, de manera que cada resultado de la iteración sea un programa aprobado por el cliente de quien depende la definición de las siguientes iteraciones.

Una característica distintiva de XP es la programación en parejas, con el objetivo de que el código sea revisado y validado antes de ser escrito; la refactorización de código está presente durante todo el desarrollo, lo cual permite reescribir el código fuente buscando claridad pero sin cambiar la funcionalidad resultante. Las parejas no serán siempre las mismas, sino que se pretende que cada desarrollador haya formado dupla al menos una vez con todos los demás, de donde se desprende que el código es de propiedad colectiva y cada uno es responsable por todo el proyecto. (15)

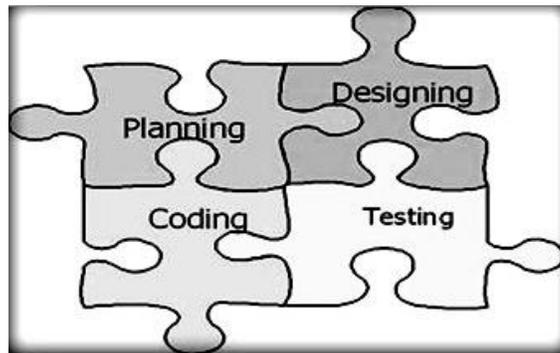


Figura 2: Metodología XP

Entre sus múltiples características, se destacan algunas relevantes que lo hace efectivo para proyectos de corto tiempo y un reducido grupo de personas como:

- XP está concebida para ser utilizada dentro de proyectos pequeños.
- Los requisitos del sistema cambian frecuentemente: Con la aceptación de nuevos requerimientos, el sistema debe cambiar y ampliar sus funcionalidades de forma que sea capaz de adaptarse a cada nueva situación. Uno de los principios básicos de XP es que el cambio frecuente de los requerimientos es algo normal en el proceso de desarrollo. Esta metodología se adapta perfectamente a los proyectos cuyos requerimientos cambian a menudo.
- El cliente forma parte del equipo de desarrollo: Mediante la aplicación de XP se puede lograr una retroalimentación mayor y lograr un producto que satisfaga sus necesidades.

- Poca disponibilidad de personal: El sistema debe ser realizado por dos personas solamente, no siendo posible la existencia de muchos roles ni la especialización en un rol específico por parte de los miembros. Uno de los principios básicos de XP es la programación en equipos pequeños (2 a 12 personas) con pocos roles, pudiendo los miembros del equipo intercambiar responsabilidades en un momento determinado.
- Propiedad colectiva del código: XP plantea que todos los programadores pueden realizar cambios en cualquier parte del código en cualquier momento.

Selección de la Metodología de Software

Como se ha analizado RUP pertenece al grupo de las metodologías robustas la cual necesita de un considerable número de personas. Se caracteriza además, por ser de vital importancia la estabilidad de los requisitos. Cada iteración está condicionada por la anterior, además, se necesita un buen líder de proyecto para garantizar el trabajo del equipo de desarrollo, existiendo una considerable cantidad de responsabilidades dentro del proyecto. En el ciclo de vida del software se realiza un gran número de artefactos, lo que puede provocar retrasos por mala preparación de los analistas. Es soportado por herramientas que automatizan, entre otras cosas, el modelo visual, la administración de cambios y las pruebas. Por el grado de complejidad, en proyectos pequeños sería tedioso realizar todas las fases en un periodo corto de tiempo, además, no se podría aplicar mucho lo de interactividad en cada fase, resultando no muy adecuado para dichos proyectos, en este caso no se ajusta para la presente trabajo. Por su parte, XP es una metodología incluida dentro de las denominadas ágiles, se encamina más a lograr entregas desde el principio basadas en la comunicación e interacción directa que debe existir entre el equipo de desarrollo y el cliente, siendo este último parte del mismo, además, existe poca disponibilidad de personal dentro del grupo de trabajo. Esta metodología se adapta a los proyectos donde los requisitos cambian frecuentemente, característica que no es ajena a la presente investigación. Analizadas las características de las anteriores metodologías de desarrollo de software, es selecciona **XP** como la candidata para el trabajo de investigación, la cual se ajusta perfectamente al mismo.

1.3.6 Framework

Un framework o marco –traducción aproximada al español - simplifica el desarrollo de aplicaciones mediante la automatización de algunos de los patrones más utilizados. Además, brinda una estructura al código fuente, obligando a los desarrolladores a escribir código legible, limpio y fácil de mantener. Por otra parte un framework facilita la programación de las aplicaciones debido a que encapsula operaciones complejas en instrucciones sencillas. (16)

En lugar de invertir los primeros días o semanas del proyecto en el diseño de la infraestructura sobre la que construir la aplicación, con el uso de un framework se puede comenzar directamente a diseñar y desarrollar los módulos que la componen, lo que supone un considerable ahorro de tiempo y permite mostrar al cliente final versiones funcionales de la aplicación muy al principio del ciclo de desarrollo.

Siempre se ha argumentado que PHP genera código difícil de mantener porque al crecer la aplicación el código se vuelve más complicado, confuso, difícil de seguir y entender. Sin embargo, una aplicación programada con la utilización de un framework para PHP no presenta estos problemas dado que aprovecha al máximo las ventajas de PHP mientras que, de forma paralela, facilita las herramientas para reducir los problemas que presentan las aplicaciones desarrolladas en este lenguaje.

Symfony

El framework symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones Web, symfony está desarrollado completamente con PHP5, por lo cual es completamente orientado a objetos. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Vale destacar que symfony es compatible con la mayoría de los sistemas gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Además, implementa la mayoría de las mejores prácticas y patrones de diseño para la programación Web.

Principales características:

- Fácil de instalar y configurar en la mayoría de plataformas.
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la Web.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con las librerías de otros fabricantes.

Zend Framework

Es un framework de código abierto para aplicaciones Web orientado a objetos desarrollado en PHP 5 y registrado bajo la nueva licencia BSD. Zend framework (a menudo denominado ZF) se desarrolla con el objetivo de simplificar el desarrollo Web y al mismo tiempo promover las mejores prácticas en la comunidad de desarrolladores de PHP.

Una de sus principales ventajas es que fue lanzado por la casa Zend, impulsora de la tecnología de servidor PHP, por lo cual se complementa de manera excelente con los entornos de desarrollo creados por la misma, sin embargo, dichos entornos no clasifican dentro de las herramientas de software libre.

Selección del Framework PHP

Tanto symfony como Zend Framework son frameworks PHP5 con gran éxito entre los desarrolladores, sin embargo, ha llegado el momento de tomar una decisión.

Dos de las ventajas más importantes con las que cuenta symfony son la calidad de su código fuente y la gran cantidad de documentación disponible. Además, symfony es una herramienta ideal para el desarrollo de aplicaciones rápidas y se adapta fácilmente a los cambios que puedan surgir durante el desarrollo de software.

Tiene también a su favor que en la UCI varias facultades cuentan con experiencia en el desarrollo de aplicaciones Web mediante su uso.

Por tanto, se selecciona **symfony** como framework PHP atendiendo, además, a las siguientes especificaciones.

Porque facilita el trabajo:

- Facilita herramientas para desarrollar aplicaciones Web de alta complejidad.
- Extensible a través de multitud de plug-ins disponibles para su descarga e instalación.
- El código desarrollado es más fácil de mantener.

Por su excepcional diseño:

- Implementación, de manera bien creativa, del patrón de diseño Modelo-Vista-Controlador.
- Utilización de otros patrones de diseño como Decorator y Separación en Capas.

Porque dispone de una extensa documentación:

- Se ha publicado un manual – Symfony, la guía definitiva - que, además de estar disponible para su consulta en línea, se actualiza periódicamente.(17)
- Gran cantidad de tutoriales y ejemplos de código disponible tanto en Internet como en la UCI.

1.3.7 Arquitectura

Arquitectura en Capas

La programación por capas es un estilo de programación en la que el objetivo primordial es la separación entre de la lógica de negocios y la lógica de diseño. La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles y en caso de algún cambio sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.

La arquitectura tres capas cuenta con una interfaz gráfica que facilita al usuario el uso del sistema (Capa 1: Capa de Presentación.), con una capa para centralizar la lógica de negocio (Capa 2: Lógica de Negocio.) y por último una capa que servirá para guardar los datos (Capa 3: Base de Datos.). A continuación se muestra la figura que presenta la arquitectura en 3 capas.

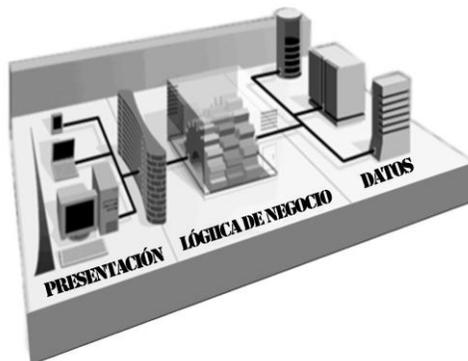


Figura 3: Arquitectura 3 Capas

Modelo-Vista-Controlador (MVC)

El Modelo-Vista-Controlador es un patrón de diseño de arquitectura que está asociado a la idea de 3 capas, aunque su objetivo es aún más fino. El mismo se centra en la secuencia de ejecución, desde que se produce un evento en la capa de presentación hasta que el mismo es atendido en forma completa.

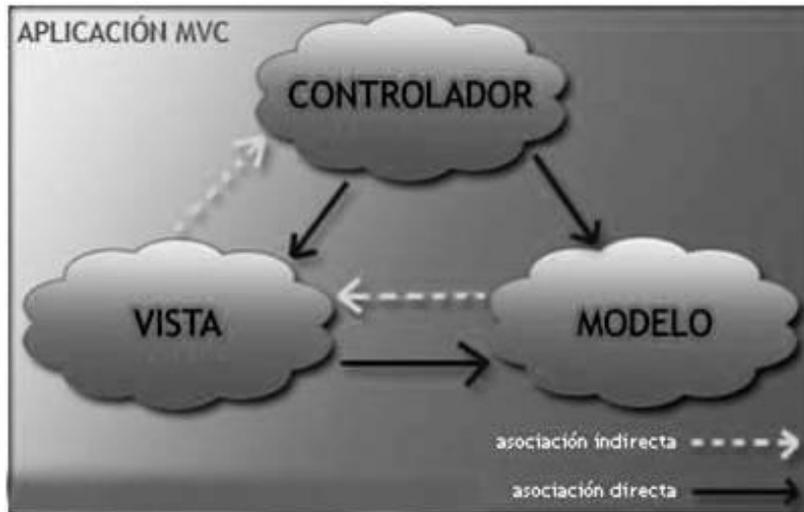


Figura 4: Arquitectura MVC

Las partes que lo componen son:

Vista: Transforma el modelo en una página web que permite al usuario interactuar con ella. Recibe el estímulo y genera un evento, que puede involucrar a otros objetos de la IU. Corresponde a la capa de presentación, y al Frontend (en Web: HTML, JavaScript,...), aunque suele haber parte en el MiddleWare (controles ASP, PHP,...)

Modelo: Representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.

Controlador: Se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. Asociado a la lógica de procesos del negocio.

Selección de la Arquitectura:

Si bien los conceptos son similares, la diferencia sería la siguiente:

- ✓ La Arquitectura MVC, es triangular es decir que hay una relación entre la vista y el controlador, entre el controlador y el modelo y entre la vista y el modelo.
- ✓ La Arquitectura a 3 Capas, es lineal es decir que no hay una comunicación directa entre las diferentes capas. Por ejemplo la capa del cliente no se comunica directamente con la capa de datos, todas las comunicaciones deben pasar por una capa intermedia. A demás que esta emerge en la década del 90, 10 años después que la Arquitectura MVC.

Como propuesta de solución se plantea el uso de la arquitectura Modelo-Vista-Controlador teniendo en cuenta las diferencias existentes entre las dos arquitecturas.

Como anteriormente se había observado las ventajas del Framework Symfony el cual utiliza la arquitectura MVC, y en la realización del software se utilizará este Framework, se selecciona la Arquitectura MVC.

1.3.8 Herramientas web utilizadas en la propuesta de solución

Aptana Studio

Se trata de un entorno de desarrollo integrado (IDE) para aplicaciones Web de software libre. Aptana está basado en el conocido entorno de desarrollo Eclipse que, de igual manera, es de software libre. Pero mientras que Eclipse está focalizado fundamentalmente en el desarrollo para Java, Aptana Studio es una distribución focalizada en el desarrollo Web, con soporte a XHTML, CSS y Javascript, así como opcionalmente a otras tecnologías como PHP y Ruby on Rails, entre otros. Se puede encontrar disponible como una aplicación independiente o como plug-in para Eclipse.

En resumen, estas son algunas de las características principales de Aptana Studio:

- Ayudas visuales para la escritura de scripts en diversos lenguajes, como coloreado y auto escritura del código, ayudas contextuales de referencia a medida que se escribe, entre otras.
- Visualización de errores de sintaxis a medida que se escribe.
- Librerías de funciones en Javascript populares en Ajax/Javascript para utilizar en los proyectos.
- Ejemplos ya creados para empezar a conocer las posibilidades de desarrollo rápidamente.
- Pre-visualización de estilos con el editor CSS. (18)

Eclipse

Es un IDE de programación gratuito que lleva tiempo demostrando su hegemonía asociado comúnmente al lenguaje de programación Java. Constituye un armazón sobre el que se pueden montar herramientas de desarrollo para cualquier lenguaje, mediante la implementación de los plug-ins adecuados. La arquitectura de plug-ins de Eclipse permite, además de integrar diversos lenguajes sobre un mismo IDE, introducir otras aplicaciones accesorias que pueden resultar útiles durante el proceso de desarrollo como: herramientas UML, editores visuales de interfaces y ayuda en línea para librerías.

Después de posicionarse como un software de desarrollo altamente competitivo, Eclipse está demostrando su calidad en el desarrollo de aplicaciones Web mediante el lenguaje PHP con el plug-in para PHP. Dicho plug-in permite a los desarrolladores en PHP un trabajo rápido, cómodo y sencillo pues tiene autocompletamiento y colorea errores. Actualmente se encuentra disponible la versión 2.1 de los PHP Development Tools (PDT) de Eclipse, los componentes especialmente diseñados para facilitar el desarrollo en PHP de acuerdo con los estándares del proyecto Eclipse. Entre otras cosas, los PDT agregados al IDE de Eclipse permiten:

- Editar tu código PHP.
- Usar asistentes de codificación.
- Usar plantillas de código PHP.
- Formatear automáticamente el código.
- Navegar por los elementos del código PHP.

PDT 2.1 incluye además, un robusto soporte de características de orientación a objetos, como herencia y polimorfismo, siendo también más pequeño y rápido gracias a la reducción de sus dependencias innecesarias (como componentes Java y J2EE). (19)

EasyEclipse

Este es un IDE que facilita el desarrollo de aplicaciones Web, su objetivo principal es la facilidad de instalación. (20)

El proyecto EasyEclipse empaqueta el entorno de desarrollo Eclipse junto con una cuidada selección de plug-ins para obtener un IDE final excepcionalmente bueno para el desarrollo de aplicaciones en PHP, Python, Ruby y por supuesto Java, con todos los plug-ins ya instalados y configurados para que el desarrollador final sólo tenga que preocuparse del código de su aplicación y no de afinar su IDE.

EasyEclipse dispone de varias "distribuciones": para desarrollo Java de servidor, de aplicaciones Java de escritorio, para dispositivos móviles, para LAMP, para PHP, para Python y para Ruby, y por supuesto, EasyEclipse es opensource y multiplataforma. (21)

EasyEclipse para PHP es un IDE muy funcional que ayuda a programar en PHP5 con funcionalidades como:

- Resaltado de código, asistente de código y autocompletado de código.

- Soporte para el debug incremental del código de PHP.
- Uso de plantillas de código PHP.
- Creación de plantillas propias además de las definidas.
- Navegación por los elementos del código PHP.

Selección de las herramientas web

En resumen, un entorno de desarrollo integrado facilita el trabajo mediante funciones tan deseadas por cualquier desarrollador como el completamiento de código o el coloreado de sintaxis. Aun cuando muchas de las ventajas antes mencionadas se han estandarizado, cada entorno de desarrollo presenta siempre un conjunto de características que hacen a los desarrolladores variar su selección entre unos u otros.

Por tanto, se selecciona **Aptana Studio** como herramienta relacionada a los lenguajes XHTML, JavaScript y CSS atendiendo al ideal soporte que posee para los mismos.

Por otro lado, se selecciona **Eclipse** como herramienta relacionada al lenguaje PHP en consecuencia con las múltiples ventajas que permite su uso, además, es un IDE muy utilizado en la Universidad de las Ciencias Informáticas.

1.4 Conclusiones

El presente capítulo constituye, sin dudas, la base sobre la cual se edificará la solución al problema de la investigación. El estudio de los sistemas de gestión de información deportiva existentes permitió identificar las características o cualidades que no deben faltar en el sistema que se va a desarrollar. Además, se pudo comprobar la necesidad del mismo atendiendo a las ventajas que brindará a sus usuarios.

También se realizó un estudio de las tendencias y tecnologías que en la actualidad se adueñan de la Industria del Software confeccionándose una propuesta de solución. Los criterios de selección que condicionaron dicha propuesta no deben considerarse para nada absolutos una vez que fueron elegidos atendiendo a las competencias del equipo de desarrolladores y a las particularidades del proceso de desarrollo.

2.1. Introducción

Es común dentro del excitante mundo del desarrollo de software que antes de comenzar un proyecto no se tengan del todo claro aspectos determinantes como el objetivo del mismo, qué se pretende alcanzar, qué tiempo demorará, entre otros. Quizás suene hasta un poco incómodo el hecho de tener que planificar el futuro de algo que es aún incierto. No obstante, dicha planificación es la única alternativa capaz de guiar de manera eficiente el trabajo futuro para conseguir la calidad deseada y por consiguiente, la aceptación de los clientes. Para llevar a cabo lo expuesto anteriormente, la metodología XP propone el desarrollo de las fases de Planificación y Exploración, las cuales, de acuerdo con las principales características del sistema a desarrollar, serán abordadas a continuación.

2.2. Flujo actual de los procesos

El desarrollo de la gestión de la información inherente a los Juegos Deportivos Inter-Facultades fluye de la siguiente manera:

En primera instancia la comisión organizadora conforma el cronograma del evento tanto general como de las diferentes competencias que conforman el mismo, este último queda plasmado en las convocatorias individuales de cada disciplina deportiva. Una vez comenzados los juegos, el puesto de mando es el encargado de tramitar todos los datos relacionados con los resultados de cada una de las competencias antes mencionadas así como las noticias e informaciones generales que se derivan de su desarrollo. Cada noche al finalizar la jornada competitiva, la comisión de emulación de los juegos tiene la responsabilidad de analizar tanto los resultados como las incidencias para finalmente otorgar la puntuación obtenida por cada facultad.

2.3. Objeto de automatización

Para llevar a cabo la gestión de la información inherente a los Juegos Deportivos Inter-Facultades existen varios procesos que deben ser automatizados pues su ejecución de forma manual resulta tediosa y muchas veces complicada, lo cual trae consigo una serie de problemas que atentan contra la organización de los mismos. Serán objeto de automatización los procesos vinculados a la visualización del programa de competencia de los Juegos Deportivos Inter-Facultades, también se mostrarán los resultados de las diferentes competencias planificadas en dicho programa. Se entiende por resultados

tanto los que se derivan de una competencia en particular como las posiciones finales de cada una de las facultades participantes en los juegos. Otro de los procesos que serán objeto de automatización será mostrar la tabla de medallas y la emulación.

2.4. Propuesta del sistema

El presente trabajo propone implementar un sistema que brinde una serie de funcionalidades, acordes al flujo de trabajo actual. Dicho sistema debe ser capaz de mostrar toda la información derivada de la creación de nuevas ediciones de Juegos Deportivos Inter-Facultades. Las competencias serán agrupadas, atendiendo a sus características, según lo especificado a continuación:

- Deportes con Pelota
- Deportes de Combate
- Modalidades
- Otros Deportes

Todas las competencias tendrán criterios de búsquedas que facilitarán el acceso a la información, siendo común la selección de la edición deportiva que se desea consultar. En el caso de las competencias que incluyen deportes con pelota se brindará la posibilidad de organizar la búsqueda utilizando los criterios como sexo, grupo deportivo y fecha, mostrando así los enfrentamientos por grupos de las distintas facultades. En el caso de las competencias relacionadas a los deportes de combate, el sistema deberá mostrar los resultados de los diferentes eventos según la división del atleta. Las competencias por modalidades, igualmente brindarán la posibilidad de organizar la búsqueda según el interés del usuario. Por último, el sistema debe brindar la posibilidad de mostrar otros tipos de competencias, estas son las que no presentan un sistema competitivo definido, por cuanto solo interesa visualizar los resultados finales de las mismas.

Para el desarrollo de los proyectos symfony propone la realización de dos aplicaciones: por un lado se encuentra la aplicación pública o “frontal” encargada de visualizar la información gestionada mientras que, del otro lado, la aplicación privada o “trasera”, tendrá como objetivo fundamental permitir la gestión de la misma. A partir del análisis de las características que el sistema en cuestión debe poseer se implementará la aplicación “frontal” del proyecto.

2.4.1. Personal relacionado con el sistema

Una de las premisas fundamentales a tener en cuenta cuando se comienza el desarrollo de cualquier sistema informático, la constituye el delimitar la audiencia a la cual va dirigido el mismo, teniendo en cuenta que esta puede estar dividida a su vez en grupos atendiendo a sus competencias. Dentro de la audiencia antes mencionada se incluyen, como personal relacionado al sistema, al administrador que será el encargado de gestionar toda la información relacionada con los juegos Inter-Facultades. A partir de la propuesta de diseño elaborada se hace necesario la presencia de un grupo: Administrador

Tabla 1. Personal relacionado con el sistema

Personal relacionado con el sistema	Descripción
Administrador	Es la persona facultada para la gestión del sistema en general. Es el encargado de actualizar diariamente la información que se encuentra en el sitio.

2.5. Fase de Exploración

Esta fase permite enmarcar el alcance del proyecto que tiene como fin el desarrollo y la entrega del sistema requerido. Para ello los clientes definen sus necesidades a través de las historias de usuario, a partir de las cuales los programadores estiman el tiempo de desarrollo. Debe quedar claro que las estimaciones realizadas en esta fase son primarias (ya que estarán basadas en datos de muy alto nivel), y podrían variar cuando se analicen más en detalle en cada iteración.

2.5.1. Historias de Usuario (HU)

Las historias de usuario son escritas por el mismo cliente, el cual en su propio lenguaje, describe lo que el sistema debe realizar. La diferencia más notable entre estas historias y los documentos de especificación de requisitos, utilizados por RUP, se encuentra en el nivel de detalle requerido. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios.

Tabla 1. HU Mostrar Resultados por Deportes

Historia de Usuario	
No. : 1	Nombre: Mostrar Resultados por Deportes.
Usuario: Administrador	
Prioridad en el Negocio: Alta	Riesgo de desarrollo : Bajo
Puntos de Estimación: 2	Iteración Asignada: 1
Descripción: El sistema debe mostrar el resultado del deporte seleccionado. En caso de ser deporte de modalidad, se mostraran los resultados por modalidad y en caso de ser deporte por división se procederá de la misma forma.	
Observación:	

Tabla 2. HU Mostrar Cronograma de Competencia

Historia de Usuario	
No. : 2	Nombre: Mostrar Cronograma de Competencia.
Usuario: Administrador	
Prioridad en el Negocio: Alta	Riesgo de desarrollo : Bajo
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: El sistema debe mostrar el cronograma de competencias de los Juegos Deportivos Inter-Facultades	
Observación:	

Tabla 3. HU Mostrar Bases

Historia de Usuario	
No. : 3	Nombre: Mostrar Bases.
Usuario: Administrador	
Prioridad en el Negocio: Alta	Riesgo de desarrollo : Bajo
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: El sistema debe mostrar las bases de cada deporte.	
Observación:	

Tabla 4. HU Mostrar Noticias

Historia de Usuario	
No. : 4	Nombre: Mostrar Noticias.
Usuario: Administrador	
Prioridad en el Negocio: Alta	Riesgo de desarrollo : Bajo
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: El sistema debe mostrar información de los hechos más relevantes acontecidos diariamente.	
Observación:	

Tabla 5. HU Mostrar Tabla de Emulación

Historia de Usuario	
No. : 5	Nombre: Mostrar Tabla de Emulación.
Usuario: Administrador	
Prioridad en el Negocio: Alta	Riesgo de desarrollo : Bajo
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: El sistema debe mostrar la puntuación acumulada diariamente para que sea de conocimiento general los lugares en los que marchan las distintas facultades.	
Observación:	

Tabla 7. HU Mostrar Medallero

Historia de Usuario	
No. : 6	Nombre: Mostrar Medallero.
Usuario: Administrador	
Prioridad en el Negocio: Alta	Riesgo de desarrollo : Bajo
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: El sistema debe mostrar la cantidad de medallas obtenidas por cada una de las distintas facultades.	
Observación:	

2.6. Fase de Planificación

En esta fase el cliente establece la prioridad de cada historia de usuario, momento a partir del cual los programadores realizan una estimación del esfuerzo necesario para desarrollar cada una de ellas. De

modo similar se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma, ambas actividades se realizan de conjunto con el cliente.

2.6.1. Estimación de esfuerzo por Historias de Usuario

Para el buen desarrollo del sistema propuesto, se realizó una estimación de cada una de las historias de usuario identificadas, la cual arrojó los resultados que se muestran a continuación:

Tabla 8. Estimación de esfuerzo por HU

Historias de usuario	Puntos de estimación
Mostrar Resultados por Deportes	2
Mostrar Cronograma de Competencia	1
Mostrar Bases	1
Mostrar Noticias	1
Mostrar Tabla de Emulación	1
Mostrar Medallero	1

2.6.2. Plan de iteraciones

Una vez descritas las historias de usuario por parte del cliente y estimado el esfuerzo por los desarrolladores para la realización de las mismas, se procede a realizar la planificación de la etapa de implementación del sistema. Este plan agrupa las historias de usuario por iteraciones, especificando cuáles serán desarrolladas en cada iteración del proceso de implementación. En relación con lo antes mencionado se decide implementar el sistema en dos iteraciones, las cuales se describen a continuación:

Iteración 1

Esta iteración tiene como objetivo darle cumplimiento a las historias de usuario 1, 2 y 3. Dichas historias de usuario son muy importantes para la aplicación. Las historias de usuario, antes mencionadas, garantizan que sea del conocimiento general las bases de cada deporte y además, muestra de manera clara y entendible el cronograma por el cual se registrarán todos los deportes, lo cual es de vital importancia

para el buen desarrollo de la edición de los Juegos Inter-Facultades en curso, además de mostrar los resultados de cada competencia que se realicen. Al finalizar esta iteración se realizará la primera entrega del sistema con el fin de mostrar al cliente lo realizado y recibir retroalimentación del mismo.

Iteración 2

Esta iteración tiene como objetivo darle cumplimiento a las historias de usuario 4, 5 y 6. La implementación de estas historias de usuario dará cumplimiento a otro grupo de funcionalidades requeridas por la aplicación. Dichas funcionalidades están estrechamente vinculadas a los siguientes procesos dentro del sistema: Mostrar la tabla de posición de la edición en curso de los juegos deportivos tanto en su variante de medallero como en la de puntuación. Las noticias serán informaciones del acontecer de los juegos Inter-Facultades que se desarrollan en ese momento.

Con la culminación de la iteración ya se tendrá la aplicación lista para, en conjunto con las funcionalidades obtenida de la anterior iteración, presentarle la primera versión del sistema completo al cliente.

2.6.3. Plan de duración de las iteraciones

Como parte del ciclo de vida de un proyecto utilizando la metodología XP se crea el plan de duración de cada una de las iteraciones. Este plan tiene como finalidad mostrar la duración de cada iteración, así como el orden en que serán implementadas las historias de usuario en cada una de ellas.

Tabla 9. Plan de duración de las iteraciones.

Iteración	Historias de Usuario	Duración total iteraciones
Iteración 1	Mostrar Resultados por Deportes	6 semanas
	Mostrar Cronograma de Competencias	
	Mostrar Bases	
Iteración 2	Mostrar Noticias	6 semanas
	Mostrar Tabla de Emulación	
	Mostrar Medallero	

2.6.4. Plan de entrega

A continuación se presenta el plan de entrega elaborado para la fase de implementación. Con el fin de facilitar la creación de dicho plan se agruparon las HU relacionadas a un mismo tema en módulos, quedando de la siguiente manera:

Tabla 10. Módulos e HU abarcadas.

Módulos	Historias de Usuario
Resultados	Mostrar Resultados por Deportes.
Programa de Competencia	Mostrar Cronograma de Competencias.
Emulación	Mostrar Medallero.
	Mostrar Tabla de Emulación.
Noticias	Mostrar Noticias.
Reseñas	Mostrar Bases.

Una vez confeccionado el plan de entrega, se hace necesario especificar que el mismo incluye una serie de “releases” del sistema los cuales irán incorporando los diferentes módulos de la aplicación, las fechas de entrega de los mencionados “releases” se indican a continuación:

Tabla 12. Plan de duración de entregas.

Módulos	Final de 1ra iteración 2da sem. abril	Final de 1ra itera. 4ta sem. abril
Resultados	0.1	1.0
Programa de Competencia	0.1	1.0
Medallas y Emulación		1.0
Noticias		1.0
Reseñas		1.0

2.7 Conclusiones

En este capítulo se comenzó a desarrollar la propuesta de solución que se desea implementar, tras el análisis del flujo de trabajo actual de los procesos descritos por el cliente. Se obtuvo, además, el conjunto de artefactos propuestos por las fases de Exploración y Planificación los cuales permitieron establecer claridad en aspectos sumamente importantes como el alcance del sistema, el tiempo estimado para dar cumplimiento a cada uno de los procesos a automatizar así como la definición de fechas “topes” de entrega de las versiones logradas del sistema a desarrollar.

3.1. Introducción

Es sabido por todos que el momento cumbre del desarrollo de un proyecto es la implementación de cada una de las funcionalidades requeridas por el mismo. Realmente reconforta ver como lo que ha sido planificado se materializa poco a poco. También es conocido que existe otro momento, estrechamente vinculado a la implementación, al cual en muchas ocasiones no se le brinda la importancia que requiere. Es el caso de las pruebas que se realizan al sistema, las cuales no deben ser vistas como trabas sino como pilares que garantizan que la implementación transite por caminos firmes. En la actualidad ambos momentos están muy entrelazados debido a que existe la tendencia de diseñar las pruebas antes que la implementación. Las características del desarrollo inherente a la implementación y prueba a las cuales se hace alusión en el presente capítulo se encuentran condicionadas a las propuestas por el framework de desarrollo utilizado.

3.2. Diseño del sistema

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC, que está formado por tres niveles: modelo, vista y controlador. Sin embargo, como se había planteado en el primer capítulo del presente trabajo, symfony implementa dicho patrón de manera bien creativa siguiendo la premisa de que la programación se puede simplificar si se utilizan otros patrones de diseño. De esta forma, las capas del modelo, la vista y el controlador se pueden subdividir en más capas. La mayoría de las aplicaciones Web suelen incluir elementos mostrados de igual manera a lo largo de toda la aplicación por lo cual la capa de la vista, asociada al patrón MVC, es más configurable si se aprovecha la separación de código. Como normalmente solo cambia la parte interna de la página, la vista se separa en layout y plantilla. Generalmente el layout se mantiene para toda una aplicación o al menos gran parte de ella mientras que la plantilla es la encargada de mostrar las variables que han sido definidas por el controlador.

En la mayoría de las aplicaciones Web la capa asociada al controlador en el patrón MVC suele realizar un trabajo excesivo. Muchas de las tareas que realiza son comunes a todos los controladores de la aplicación como el manejo de las peticiones del usuario, el manejo de la seguridad, cargar la configuración de la aplicación, entre otras. Ese es el motivo por el cual el controlador se divide en un controlador frontal, único para toda la aplicación, y las acciones, que implementan el código específico del controlador de cada página. Por otro lado, la lógica del negocio en este tipo de aplicaciones depende casi en su totalidad de su modelo de datos. Precisamente la capa del modelo puede separarse en la

capa de acceso a los datos y en la capa de abstracción de la base de datos con lo cual se logra que el acceso a datos no dependa de ningún sistema gestor de base de datos en particular.

3.2.1. Tarjetas CRC

Sin dudas una de las tareas más importantes en cuanto a diseño de una aplicación se refiere la constituye el definir correctamente las clases que contendrán la lógica del negocio. La metodología XP no obliga a la realización de diagramas UML para la presentación de las mismas y en su lugar propone el uso de tarjetas CRC (Clase-Responsabilidad-Colaboración), técnica que ayuda a evitar el enfoque procedimental destacando la orientación a objetos.

Tabla 13. Tarjeta CRC clase Competencia.

Competencia	
Funcionalidades	Colaboración
Devolver el valor de un atributo.	Grupo
Devolver el deporte al que pertenece la competencia.	Evento
Devolver los eventos que pertenecen a la competencia.	Deporte
Devolver los grupos que pertenecen a la competencia.	
Devolver los equipos disponibles para una competencia.	
Devolver los eventos que pertenecen a la competencia.	

Tabla 14. Tarjeta CRC clase Deporte.

Deporte	
Funcionalidades	Colaboración
Devolver el valor de un atributo.	Competencia
Devolver las competencias asociadas al deporte.	Equipo
Devolver los equipos asociados al deporte.	Modalidad
Devolver las modalidades asociadas al deporte.	

Tabla 15. Tarjeta CRC clase Enfrentamiento.

Enfrentamiento	
Funcionalidades	Colaboración
Devolver el valor de un atributo.	Evento
Devolver el resultado de un enfrentamiento.	Equipo
Devolver los equipos asociados al deporte.	
Devolver las modalidades asociadas al deporte.	

Tabla 16. Tarjeta CRC clase Equipo.

Equipo	
Funcionalidades	Colaboración
Devolver el valor de un atributo.	Deporte
Devolver el deporte al que pertenece el equipo.	Grupo
Devolver el grupo al que pertenece el equipo.	JuegoFacultad
Conocer la posición final de un equipo en una competencia.	
Devolver la facultad a la que pertenece el equipo.	
Devolver los equipos disponibles para una competencia.	

Tabla 16. Tarjeta CRC clase Evento.

Evento	
Funcionalidades	Colaboración
Devolver el valor de un atributo.	Competencia
Devolver la competencia a la que pertenece el evento.	EventoModalidad
Devolver el enfrentamiento asociado a este evento.	Enfrentamiento
Devolver el evento por modalidad asociado a este evento.	
Devolver los atletas que han sido ganadores en un evento.	

Tabla 17. Tarjeta CRC clase EventoModalidad.

EventoModalidad	
Funcionalidades	Colaboración
Devolver el valor de un atributo.	Evento
Devolver el evento por modalidad.	Modalidad

Tabla 18. Tarjeta CRC clase EventoDivisión.

EventoDivisión	
Funcionalidades	Colaboración
Devolver el valor de un atributo.	Evento
Devolver el evento por división.	División

Tabla 19. Tarjeta CRC clase EventoDivisión.

EventoDivisión	
Funcionalidades	Colaboración
Devolver el valor de un atributo.	JuegoFacultad
Devolver los juegos donde estuvo presente la facultad.	

Tabla 19. Tarjeta CRC clase Grupo.

Grupo	
Funcionalidades	Colaboración
Devolver el valor de un atributo.	Equipo
Devolver la competencia a la que pertenece este grupo.	Competencia
Devolver los equipos que pertenecen a este grupo.	
Devolver los equipos que no estén en grupo.	

Tabla 20. Tarjeta CRC clase Juego.

Juego	
Funcionalidades	Colaboración
Devolver el valor de un atributo.	JuegoFacultad
Devolver las competencias asociadas a esta edición de juego.	Competencia
Devolver las facultades presentes en esta edición ordenadas por su puntuación.	
Devolver las facultades presentes en esta edición ordenadas por sus medallas.	

Tabla 21. Tarjeta CRC clase JuegoFacultad.

JuegoFacultad	
Funcionalidades	Colaboración
Devolver el valor de un atributo.	Juego
Devolver los equipos de una facultad asociados a una edición de juego.	Facultad
Devolver las medallas obtenidas por una facultad en una edición de juego.	Equipo
Conocer la posición final de una delegación en una edición de juego.	
Devolver la puntuación obtenida por una facultad en una edición de juego.	

Tabla 22. Tarjeta CRC clase Modalidad.

Modalidad	
Funcionalidades	Colaboración
Devolver el valor de un atributo.	Deporte
Devolver el deporte de la modalidad.	EventoModalidad
Devolver los eventos por modalidad a los que está asociado esta modalidad.	

Tabla 23. Tarjeta CRC clase División.

División	
Funcionalidades	Colaboración
Devolver el valor de un atributo.	Deporte
Devolver el deporte de la división.	EventoDivisión
Devolver los eventos por división a los que está asociado esta división.	

Tabla 24. Tarjeta CRC clase Atleta.

Atleta	
Funcionalidades	Colaboración
Devolver el valor de un atributo.	AtletaEvento
Devolver atleta ganador de un evento determinado.	

Tabla 25. Tarjeta CRC clase AtletaEvento.

AtletaEvento	
Funcionalidades	Colaboración
Devolver el valor de un atributo.	Atleta
	Evento

3.2.2. Modelo de datos

Entre los errores más graves que cometen algunos desarrolladores a la hora de realizar una aplicación Web está el no diseñar correctamente la base de datos que la soportará. Otro detalle es que mientras que symfony y PHP5 son orientados a objetos, las bases de datos son relacionales por lo cual es evidente la necesidad de crear una interfaz que “mapee” la lógica de objetos a la lógica relacional. La interfaz antes mencionada es conocida como ORM (mapeo de objetos a bases de datos) y symfony incorpora por defecto el ORM realizado por el proyecto Propel con lo cual se logra que el acceso a las bases de datos en las aplicaciones desarrolladas mediante su uso se realice a través de objetos.

Tabla 26. Transformación ORM.

Relacional	Orientado a Objetos
Tabla	Clase
Fila	Objeto
Campo	Atributo

Uno de los mayores inconvenientes del uso de un ORM es que se debe definir la estructura del modelo dos veces: una para la base de datos y otra para el modelo de objetos, no obstante symfony dispone de utilidades para generar uno en función del otro. A continuación se presenta un esbozo del modelo de datos.

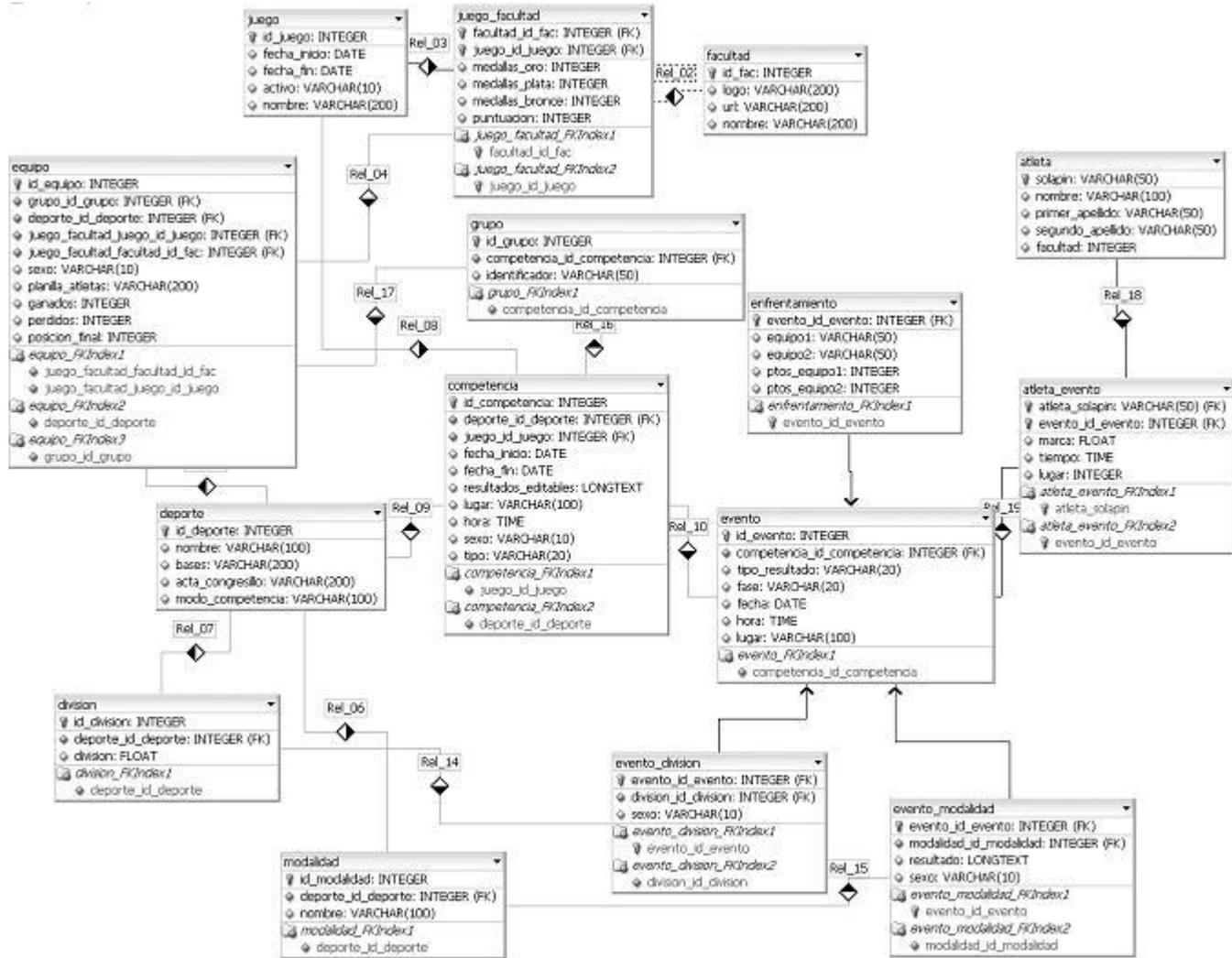


Figura 5: Modelo de Datos

3.3. Implementación

Para llevar a cabo la correcta implementación de los módulos definidos en la propuesta de solución, con las correspondientes HU que se incluyen en cada uno de ellos, se deben definir por parte del equipo de desarrollo las tareas que serán llevadas a cabo. Lo anterior permite a los desarrolladores obtener un nivel de detalle más avanzado que el sugerido por las HU. A continuación se describen las tareas específicas para el desarrollo de la presente aplicación surgidas a partir de la técnica denominada “tormenta de ideas”.

3.3.1. Iteración 1

En esta iteración se implementaron las historias de usuario que conforman la estructura básica del sistema. A partir de ellas se realizaron las demás funcionalidades requeridas por la aplicación.

Tabla 27. Módulos abordados en la primera iteración.

Módulo	Historia de usuario	Tiempo de Implementación (semana)	
		Estimado	Real
Resultados	Mostrar Resultados por Deportes	1	0.50
Cronograma de Competencia	Mostrar Cronograma por Deporte	1	0.50

A continuación se muestran las tareas efectuadas para cada una de las HU presentes en los módulos implementados en esta iteración:

Módulo Resultados

Tabla 28. Tarea 1 módulo Resultados.

Tarea	
Numero de tarea :1	Numero de HU:1
Nombre de Tarea: Crear módulo Resultados	
Tipo de tarea: Configuración	Punto de estimación: 0.50
Fecha de inicio : 1 de abril del 2010	Fecha de fin : 3 de abril del 2010
Programador responsable: Julio A. Kelly Naranjo – Julio C. Cañabate Gutiérrez	
Descripción: Se crea el módulo Resultado, atendiendo a la tecnología usada (framework symfony).	

Tabla 29. Tarea 2 módulo Resultados.

Tarea	
Numero de tarea :2	Numero de HU:1
Nombre de Tarea: Definir e implementar las acciones dentro del módulo para mostrar los resultados de deportes con pelota.	
Tipo de tarea: Desarrollo	Punto de estimación: 0.50
Fecha de inicio : 1 de abril del 2010	Fecha de fin : 3 de abril del 2010
Programador responsable: Julio A. Kelly Naranjo – Julio C. Cañabate Gutiérrez	
Descripción: Se definen e implementan las acciones que se realizarán dentro del módulo Resultados. Aquí se implementarán las acciones para mostrar los resultados de un deporte con pelota.	

Tabla 30. Tarea 3 módulo Resultados.

Tarea	
Numero de tarea :3	Numero de HU:1
Nombre de Tarea: Definir e implementar las acciones dentro del módulo para seleccionar las modalidades y divisiones.	
Tipo de tarea: Desarrollo	Punto de estimación: 0.50
Fecha de inicio : 1 de abril del 2010	Fecha de fin : 3 de abril del 2010
Programador responsable: Julio A. Kelly Naranjo – Julio C. Cañabate Gutiérrez	
Descripción: Se definen las acciones que se realizarán dentro del módulo Resultados relacionadas con las modalidades y divisiones pertenecientes a un deporte. Aquí se definen e implementan las acciones para mostrar los resultados de un deporte dada una modalidad o una división.	

Tabla 31. Tarea 4 módulo Resultados.

Tarea	
Numero de tarea :2	Numero de HU:1
Nombre de Tarea: Definir e implementar las acciones dentro del módulo para mostrar los resultados de otros deportes.	
Tipo de tarea: Desarrollo	Punto de estimación: 0.50
Fecha de inicio : 1 de abril del 2010	Fecha de fin : 3 de abril del 2010
Programador responsable: Julio A. Kelly Naranjo – Julio C. Cañabate Gutiérrez	
Descripción: Se definen e implementan las acciones que se realizarán dentro del módulo Resultados. Aquí se implementarán las acciones para mostrar los resultados de otros deportes.	

Tabla 32. Tarea 5 módulo Resultados.

Tarea	
Numero de tarea :4	Numero de HU:1
Nombre de Tarea: Diseñar las plantillas para cada acción del módulo Resultados	
Tipo de tarea: Desarrollo	Punto de estimación: 0.50
Fecha de inicio : 4 de abril del 2010	Fecha de fin : 10 de abril del 2010
Programador responsable: Julio A. Kelly Naranjo – Julio C. Cañabate Gutiérrez	
Descripción: Se diseña las plantillas para cada una de las acciones del módulo Resultados.	

Módulo Cronograma de Competencia

Tabla 33. Tarea 1 módulo Cronograma de Competencia.

Tarea	
Numero de tarea :1	Numero de HU:2
Nombre de Tarea: Crear módulo Cronograma de Competencia	
Tipo de tarea: Configuración	Punto de estimación:0.50
Fecha de inicio : 10 de abril del 2010	Fecha de fin : 14 de abril del 2010
Programador responsable: Julio A. Kelly Naranjo – Julio C. Cañabate Gutiérrez	
Descripción: Se crea el módulo Cronograma de Competencia, atendiendo a la tecnología usada (framework symfony)	

Tabla 34. Tarea 2 módulo Cronograma de Competencia.

Tarea	
Numero de tarea :2	Numero de HU:2
Nombre de Tarea: Definir e implementar las acciones dentro del módulo para mostrar el cronograma de competencia	
Tipo de tarea: Desarrollo	Punto de estimación:0.50
Fecha de inicio : 10 de abril del 2010	Fecha de fin : 14 de abril del 2010
Programador responsable: Julio A. Kelly Naranjo – Julio C. Cañabate Gutiérrez	
Descripción: Se definen e implementan las acciones que se realizarán dentro del módulo Cronograma de Competencia. Aquí se implementarán las acciones mostrar el cronograma de cada deporte.	

Tabla 35. Tarea 3 módulo Cronograma de Competencia.

Tarea	
Numero de tarea :3	Numero de HU:2
Nombre de Tarea: Diseñar las plantillas para cada acción del módulo Resultados.	
Tipo de tarea: Desarrollo	Punto de estimación: 0.50
Fecha de inicio: 14 de abril del 2010.	Fecha de fin: 15 de abril del 2010.
Programador responsable: Julio A. Kelly Naranjo – Julio C. Cañabate Gutiérrez.	
Descripción: Se diseña las plantillas para cada una de las acciones del módulo Cronograma de Competencia.	

3.3.2. Iteración 2

Durante el desarrollo de esta iteración se implementaron las historias de usuario que constituyeron las funcionalidades principales dentro de la aplicación.

Tabla 36. Módulos abordados en la segunda iteración.

Módulo	Historia de usuario	Tiempo de Implementación (semana)	
		Estimado	Real
Emulación y Medallas	Mostrar Emulación y Medallero	1	0.50
Bases	Mostrar Bases por Deportes	1	0.50

Módulo Emulación

Tabla 37. Tarea 1 módulo Emulación.

Tarea	
Numero de tarea : 1	Numero de HU: 5,6
Nombre de Tarea: Crear módulo Emulación y Medallas.	
Tipo de tarea: Configuración	Punto de estimación: 0.50
Fecha de inicio: 16 de abril del 2010.	Fecha de fin: 16 de abril del 2010.
Programador responsable: Julio A. Kelly Naranjo – Julio C. Cañabate Gutiérrez.	
Descripción: Se crea el módulo Emulación y el Medallero, atendiendo a la tecnología usada (framework symfony).	

Tabla 38. Tarea 2 módulo Emulación.

Tarea	
Numero de tarea : 2	Numero de HU: 5,6
Nombre de Tarea: Definir e implementar las acciones dentro del módulo para mostrar la emulación.	
Tipo de tarea: Desarrollo	Punto de estimación: 0.50
Fecha de inicio: 17 de abril del 2010.	Fecha de fin: 17 de abril del 2010.
Programador responsable: Julio A. Kelly Naranjo – Julio C. Cañabate Gutiérrez.	
Descripción: Se definen e implementan las acciones que se realizarán dentro del módulo Emulación. Aquí se implementarán las acciones mostrar la emulación y el medallero de cada facultad.	

Tabla 39. Tarea 3 módulo Emulación.

Tarea	
Numero de tarea :3	Numero de HU:5,6
Nombre de Tarea: Diseñar las plantillas para cada acción del módulo Resultados	
Tipo de tarea: Desarrollo	Punto de estimación: 0.50
Fecha de inicio: 17 de abril del 2010.	Fecha de fin: 25 de abril del 2010.
Programador responsable: Julio A. Kelly Naranjo – Julio C. Cañabate Gutiérrez.	
Descripción: Se diseña las plantillas para cada una de las acciones del módulo Emulación.	

Módulo Bases

Tabla 40. Tarea 1 módulo Bases.

Tarea	
Numero de tarea :1	Numero de HU:3
Nombre de Tarea: Crear módulo Bases	
Tipo de tarea: Configuración	Punto de estimación: 0.50
Fecha de inicio : 26 de abril del 2010	Fecha de fin : 26 de abril del 2010
Programador responsable: Julio A. Kelly Naranjo – Julio C. Cañabate Gutiérrez.	
Descripción: Se crea el módulo Bases, atendiendo a la tecnología usada (framework symfony).	

Tabla 41. Tarea 2 módulo Bases.

Tarea	
Numero de tarea :2	Numero de HU:3
Nombre de Tarea: Definir e implementar las acciones dentro del módulo para mostrar las bases	
Tipo de tarea: Desarrollo	Punto de estimación: 0.50
Fecha de inicio: 26 de abril del 2010.	Fecha de fin: 28 de abril del 2010.
Programador responsable: Julio A. Kelly Naranjo – Julio C. Cañabate Gutiérrez.	
Descripción: Se definen e implementan las acciones que se realizarán dentro del módulo Bases. Aquí se implementarán las acciones mostrar las bases de cada deporte.	

Tabla 42. Tarea 3 módulo Bases.

Tarea	
Numero de tarea :3	Numero de HU:3
Nombre de Tarea: Diseñar las plantillas para cada acción del módulo Bases.	
Tipo de tarea: Desarrollo.	Punto de estimación: 0.50
Fecha de inicio: 28 de abril del 2010.	Fecha de fin: 28 de abril del 2010.
Programador responsable: Julio A. Kelly Naranjo – Julio C. Cañabate Gutiérrez.	
Descripción: Se diseña las plantillas para cada una de las acciones del módulo Bases.	

3.4. Prueba

Uno de los pilares de la eXtreme Programming es el proceso de pruebas. XP anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y

su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones. XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final. A tono con la metodología XP, symfony propone la realización de pruebas con el fin de asegurar la calidad de la futura aplicación. No obstante, y aún cuando la esencia continúa siendo la misma, las pruebas recomendadas por symfony son pruebas automatizadas. Este tipo de pruebas, constituyen uno de los mayores avances en la programación desde la aparición de la orientación a objetos y aunque son recomendadas, su uso o no, queda a merced de los desarrolladores.

3.4.1. Pruebas de aceptación

Las pruebas funcionales son la mejor forma de probar la aplicación de extremo a extremo: desde la petición realizada por un navegador hasta la respuesta enviada por el servidor. Las pruebas funcionales prueban todas las capas de la aplicación: el sistema de enrutamiento, el modelo, las acciones y las plantillas. En realidad, son muy similares a lo que se hace manualmente cada vez que se añade o modifica una acción y se prueban dichos cambios en el navegador para comprobar que todo funciona bien al pulsar sobre los enlaces y botones y que todos los elementos se muestran correctamente en la página. En otras palabras, lo que se hace es probar un escenario correspondiente a la historia de usuario que se acaba de implementar en la aplicación.

Tabla 42. Prueba 1 del módulo Resultados.

Caso de prueba de aceptación	
Código: HU1_P1	Historia de Usuario: 1
Nombre: Mostrar los grupos de un deporte por el sistema.	
Descripción: Prueba para la funcionalidad de mostrar los grupos de un deporte por el sistema.	
Condiciones de Ejecución: El usuario debe seleccionar el deporte. El usuario debe seleccionar una edición de los juegos inter-facultades.	
Resultados esperados: El sistema muestra la distribución de los grupos en la edición seleccionada.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 43. Prueba 2 del módulo Resultados.

Caso de prueba de aceptación	
Código: HU1_P2	Historia de Usuario: 1
Nombre: Mostrar los resultados de un deporte por el sistema.	
Descripción: Prueba para la funcionalidad de mostrar los resultados de un deporte por el sistema.	
Condiciones de Ejecución: El usuario debe seleccionar el sexo del cual quiere ver los resultados. El usuario debe seleccionar el grupo del cual quiere ver los resultados. El usuario debe seleccionar la fecha de la cual quiere ver los resultados.	
Resultados esperados: El sistema muestra los resultados del deporte seleccionado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 44. Prueba 1 del módulo Cronograma de Competencia.

Caso de prueba de aceptación	
Código: HU2_P1	Historia de Usuario: 2
Nombre: Mostrar el cronograma de competencia por el sistema.	
Descripción: Prueba para la funcionalidad de mostrar el cronograma de competencia por el sistema.	
Condiciones de Ejecución: El usuario debe seleccionar una edición de los juegos inter-facultades. El usuario debe seleccionar el Tipo de Competencia. El usuario debe seleccionar el deporte.	
Resultados esperados: El sistema muestra el cronograma de competencia de la edición y el deporte seleccionados.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 45. Prueba 1 del módulo Bases.

Caso de prueba de aceptación	
Código: HU3_P1	Historia de Usuario: 3
Nombre: Descargar las bases de un deporte por el sistema	
Descripción: Prueba para la funcionalidad de descargar las bases de un deporte por el sistema.	
Condiciones de Ejecución: El usuario debe seleccionar el deporte.	
Resultados esperados: El sistema permite descargar las bases de la edición seleccionada.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 46. Prueba 2 del módulo Bases.

Caso de prueba de aceptación	
Código: HU3_P2	Historia de Usuario: 3
Nombre: Descargar el acta del congresillo de un deporte por el sistema.	
Descripción: Prueba para la funcionalidad de descargar el acta del congresillo de un deporte por el sistema.	
Condiciones de Ejecución: El usuario debe seleccionar el deporte.	
Resultados esperados: El sistema permite descargar el acta del congresillo de la edición seleccionada.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 47. Prueba 1 del módulo Medallero.

Caso de prueba de aceptación	
Código: HU6_P1	Historia de Usuario: 6
Nombre: Mostrar el medallero por el sistema.	
Descripción: Prueba para la funcionalidad de mostrar el medallero por el sistema.	
Condiciones de Ejecución: El usuario debe seleccionar una edición de los juegos inter-facultades.	
Resultados esperados: El sistema muestra el medallero de la edición seleccionada.	
Evaluación de la prueba: Prueba satisfactoria.	

3.5. Conclusiones

Con la elaboración del presente capítulo se obtuvo la propuesta de diseño del sistema. Se describieron las tarjetas CRC, las cuales brindan claridad en aspectos como las principales funcionalidades que presentan las clases así como la relación existente entre ellas. Por otro lado, se diseñó el modelo de datos encargado de brindar soporte a la aplicación en cuestión. Otro de los artefactos generados fue las tareas por historia de usuarios, el mismo es de vital importancia por cuanto ayudan a los desarrolladores a profundizar más en los aspectos a tener en cuenta a la hora de implementar las HU detectadas. Por último, se describió el proceso de prueba, uno de los más importantes para garantizar el éxito de la aplicación.

Con el fin del presente capítulo se da por terminada la propuesta de solución relacionada al desarrollo de la aplicación inherente al presente trabajo de diploma: "Sistema para la Gestión de la Información de los Juegos Deportivos Inter-Facultades Versión 2.0".

Conclusiones

Con el fin de este trabajo también llega a su fin un largo camino que para nada ha sido fácil. Este trabajo representa además el comienzo de una nueva etapa donde la puesta en práctica del conocimiento adquirido ha sido lo más importante. Quedaron muchas cosas por aprender, aunque se tiene la seguridad que los conocimientos adquiridos son suficientes para el nuevo periodo que se avecina.

- Se realizó un estudio de toda la información inherente al acontecer de los Juegos Deportivos Inter-Facultades lo cual permitió un mejor entendimiento de los procesos relacionados con la gestión de la información de dichos juegos.
- Se realizó la exploración y planificación de un subsistema que gestione la información de los Juegos Deportivos Inter-Facultades en la UCI lo que propició un punto de partida para la posterior implementación del subsistema.
- Se Implementó un subsistema que gestione la información de los Juegos Deportivos Inter-Facultades en la UCI eliminando así el problema de la desinformación existente en la comunidad universitaria.

Recomendaciones

- Aplicar el software en los próximos juegos deportivos con el fin de verificar su viabilidad e incorporar las correcciones pertinentes en caso de que sea necesario.
- Continuar el trabajo con el objetivo de incorporar al software nuevas funcionalidades.

Referencias Bibliográficas

1. Lic. Lourdes Aja Quiroga: Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones: [citado enero 2010]; Aparece en :
http://www.bvs.sld.cu/revistas/aci/vol10_5_02/aci04502.htm
2. Aplicación Web [citado enero 2010]; Aparece en: http://es.wikipedia.org/wiki/Aplicacion_web
3. Aplicaciones Web vs. Escritorio [citado enero 2010]; Aparece en:
<http://alxplus.blogspot.com/2006/08/aplicaciones-web-vs.html>
4. HTML y XHTML. [citado enero 2010]; Aparece en Teleformación:
<http://eva.uci.cu/mod/resource/view.php?id=20077>
5. CSS. [citado enero 2010]; Aparece en: <http://www.librosweb.es/css/capitulo1.html>
6. PERL. [citado enero 2010]; Aparece en: <http://www.desarrolloweb.com/articulos/541.php>
7. Desarrollo Web [citado enero 2010]; Aparece en:
<http://www.maestrosdelweb.com/editorial/phpintro>
8. **Castillo, A.** Características de PHP [citado enero 2010]; Aparece en:
http://www.programemos.com/index.php?option=com_content&task=view&id=279&Itemid=223
9. MySQL [citado enero 2010]; Aparece en: <http://es.wikipedia.org/wiki/MySQL>
10. Apache. [citado febrero 2010]; Aparece en <http://www.apache.or/>
11. Zope. [citado enero 2010]; Aparece en: <http://es.wikipedia.org/wiki/Zope>.
12. Introducción a la Ingeniería de Software. [citado mayo 2010]; Aparece en:
http://eva.uci.cu/mod/resource/view.php?id=2241&subdir=/Conferencias_IS1_2007-2008
13. Introducción a la Ingeniería de Software. [citado mayo 2010]; Aparece en:
http://eva.uci.cu/mod/resource/view.php?id=2241&subdir=/Conferencias_IS1_2007-2008
14. Introducción a la Ingeniería de Software. [citado mayo 2010]; Aparece en:
http://eva.uci.cu/mod/resource/view.php?id=2241&subdir=/Conferencias_IS1_2007-2008

15. **Penadés, Patricio Letelier y M^a Carmen.** *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. España: Universidad Politécnica de Valencia, 2006.
16. Maestros de la Web. Los Framework agilizan tu trabajo. Aparece en:
<http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo/>
17. symfony la guía definitiva [citado febrero 2010]; Aparece en:
http://www.librosweb.es/symfony_1_2
18. Aptana Studio [citado febrero 2010]; Aparece en: <http://www.desarrolloweb.com/articulos/aptana-studio.html>
19. Viva PHP! Eclipse PDT 2.1 [citado febrero 2010]; Aparece en:
<http://translate.google.com/cu/translate?hl=es&sl=en&u=http://www.eclipse.org/pdt/&ei=tOuBS5e5GsW2IAee>
20. Versión traducida [citado febrero 2010]; Aparece en: <http://easyeclipse.org/>
21. Viva Linux! EasyEclipse: la manera fácil de instalar Eclipse [citado febrero 2010]; Aparece en:
<http://www.vivalinux.com.ar/soft/easy-eclipse>

Bibliografía

1. **Crispin, L. y House, T.** Testing Extreme Programming. s.l.: Addison Wesley, 2002.
2. **Beck, K.** Extreme Programming Explained. s.l. : Addison Wesley, 2000
3. **Beck, K. y Fowler, M.** Planeando en Programación Extrema. 2000.
4. XP. Introduction.2006 [citado enero 2010]; Disponible en:
<http://www.extremeprogramming.org>.
5. Qué es Javascript [citado enero 2010]; Aparece en:
<http://www.desarrolloweb.com/articulos/25.php>
6. Guía Breve de CSS [citado enero 2010]; Aparece en:
<http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>

7. **EGUÍLUZ, Javier**. Introducción a CSS. 2008 [citado enero 2010]; Aparece en: <http://librosweb.es/css/>
8. Qué es Python [citado enero 2010]; Aparece en: <http://www.desarrolloweb.com/articulos/1325.php>
9. Qué es Perl [citado enero 2010]; Aparece en: <http://www.desarrolloweb.com/articulos/541.php>
10. Desarrolloweb.com. "Programación en PHP" [citado enero 2010]; Aparece en: <http://www.desarrolloweb.com/manuales/>
11. Maestros del Web Los Frameworks de PHP [citado febrero 2010]; Aparece en: <http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo/>
12. Manual de PHP 5 [citado enero 2010]; Aparece en: <http://www.desarrolloweb.com/manuales/58/>
13. Sitio oficial de PHP [citado enero 2010]; Aparece en: <http://www.php.net/>
14. **Diego Samir Melo Solarte** Base datos con software libre [citado enero 2010]; Aparece en: www.umanizales.edu.co/programs/ingenieria/ventana/ventana12/articulo25.pdf
15. **KABIR, M. J.** La biblia de Servidor Apache 2, 2003, Vol: I.
16. **Fabien Potencier, François Zaninotto**. symfony la guía definitiva 2008 [citado febrero 2010]; Aparece en: http://www.librosweb.es/symfony_1_2
17. **Fabien Potencier**. El Tutorial Jobeet [citado febrero 2010]; Aparece en: <http://www.librosweb.es/jobeeet/>
18. Sitio Web de symfony [citado febrero 2010]; Aparece en: <http://www.symfony-project.org/>
19. Lista de correos de usuarios de symfony [citado febrero 2010]; Aparece en: <http://groups.google.es/group/symfony-es>
20. El wiki de symfony [citado febrero 2010]; Aparece en: <http://trac.symfony-project.org/>