



Universidad de las Ciencias Informáticas

*Sistema para la Gestión de la Información de los Juegos
Deportivos Interfacultades.*

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

**Autores: Elizabet Rodríguez Cid
Michel Legrá Fleitas**

**Tutor: Ing. Leonardo San Roman Labaut
Co- tutor: Lic. Alexander Sedeño Puig**

**Ciudad de La Habana
Mayo de 2009**

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) a que haga el uso que estimen pertinente con el mismo.

Para que así conste firmamos la presente a los ___ días del mes de Junio del 2009.

Autores:

Elizabet Rodríguez Cid

Michel Legrá Fleitas

Tutor:

Ing.: Leonardo San Román Labaut

Co-tutor:

Lic. Alexander Sedeño Puig

“La virtud, como el arte, se consagra constantemente a lo que es difícil de hacer, y cuanto más dura es la tarea más brillante es el éxito.”

Aristóteles.

Agradecimientos

A nuestro Comandante en Jefe y a la Revolución Cubana por habernos dado la oportunidad de estudiar en la Universidad de las Ciencias Informáticas donde hicimos realidad nuestro más grande sueño.

A nuestros padres por existir pues, con su ejemplo, labraron el camino de nuestro éxito.

A nuestra familia por estar siempre pendiente de nuestro sacrificio y brindarnos su aliento.

A todos los que brindaron su ayuda para la realización de este trabajo, en especial a Raymel y a Roly.

A nuestro tutor y co-tutor por el apoyo aportado en la realización del trabajo.

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

Gracias

Dedicatoria

Elizabeth

A mi abuelin a quien le agradezco todo lo que ha hecho por mí, por estar siempre a mi lado y por lo orgullosa que se siente en estos momentos.

A mi mamá por guiarme en todo momento, apoyarme y comprender todas las decisiones que he tomado siempre.

A mi hermana por estar siempre ahí cuando la he necesitado y a la cual, a pesar de sus locuras, admiro muchísimo.

A mi papá por impulsarme cada vez a ser mejor, a esforzarme y a dar el máximo para superarme.

A mi novio y compañero de tesis por estar siempre a mi lado en los momentos buenos y malos, por el cual siento una profunda admiración.

A mi abuelo Ángel el cual estaría muy orgulloso de mí si me viera en estos momentos.

A toda mi familia que siempre ha estado pendiente de mí.

A mis compañeros que han estado los 5 años a mi lado y en especial a Mery, Yanelis, Héctor, Yan, Nelson.

Michel

Al la memoria de mi abuelo, quien fuera partícipe del inicio de un sueño que hoy se convierte en realidad.

Al mi abuela, por hacer honor al nombre que lleva y en consecuencia ser la flor de mi vida.

Al mi madre, por haberme mostrado el camino dándome las fuerzas para emprenderlo y por enseñarme que el éxito no radica en la distancia recorrida sino en caminar dejando en cada paso vida, corazón y obra, sobre todas las cosas, obra.

Al mi novia y compañera de tesis por demostrarme que no solo es excelente como pareja sino también como profesional. Gran parte del éxito del presente trabajo se lo debo a ella.

Al mi hermana mayor, quien constituye mi ejemplo, porque tal pareciera que estoy predestinado a seguir sus pasos y como siempre le tocó llegar primero, cuando fue mi turno el camino era más claro.

Al mi padre, a quien respeto y admiro, por su consejo de levantarme y trabajar cada día más duro.

Al mis tíos y primos, a los cuales agradezco los momentos compartidos.

Al mi familia en general, los que están y los que no, pues se lo orgullosos que se sienten en estos momentos.

Al mis compañeros todos, gracias por su ayuda y en especial a aquellos que estudiaron más cerca cuando reímos y cuando lloramos Israel, Nelson, Karel y Yan.

Resumen

En la actualidad la Industria del Software en Cuba está llamada a convertirse uno de los sectores más importantes, dinámicos y que aporte más beneficios a nuestra sociedad. La Universidad de las Ciencias Informáticas (UCI) ha hecho suyo el llamado creando una infraestructura productiva que brinda resultados palpables al país. En sus calles, aulas y laboratorios miles de jóvenes cada día se introducen en el excitante mundo de la producción, respondiendo así al llamado que se les ha hecho y a la confianza depositada en ellos por la Revolución. Pero estos jóvenes no son máquinas, motivo por el cual la UCI, como todas las universidades del país, fomenta su formación integral creando los diferentes espacios para lograrlo. Precisamente uno de los momentos más representativos de la universidad, lo constituyen los Juegos Deportivos Interfacultades, denominados Juegos Mella. Sin embargo la UCI no cuenta con un software que gestione la información inherente a dicho evento deportivo por demás de gran magnitud. Por tal motivo surge la necesidad de encontrar una propuesta de solución que permita mejorar cada una de las actividades realizadas en estos juegos. Para lograrlo se analiza el flujo actual de los procesos presentes en la conformación de los Juegos Mella, los cuales poseen un grupo de características que permiten incluirlos dentro de los denominados eventos deportivos múltiples. A partir del análisis anterior se identifican los procesos candidatos a automatizar para finalmente conformar la propuesta de solución que garantice obtener el resultado esperado. Dicha propuesta se basa en la elaboración de un sistema informático desarrollado utilizando las denominadas tecnologías Web, cuyo título será: “Sistema para la Gestión de la Información de los Juegos Deportivos Interfacultades”.

Índice General

Introducción	1
Capítulo 1 Fundamentación teórica	5
1.1 Introducción	5
1.2 Fundamentación del tema	5
1.2.1 Conceptos fundamentales.....	5
1.2.2 Sistemas informáticos similares	7
1.3 Tendencias y tecnologías actuales	8
1.3.1 Política de migración al Software Libre	8
1.3.2 Tipos de aplicaciones.....	9
1.3.3 Lenguajes de programación Web.....	10
1.3.4 AJAX.....	14
1.3.5 Sistemas Gestores de Base de Datos.....	15
1.3.6 Servidores Web	17
1.3.7 Metodologías de desarrollo de software.....	19
1.3.8 Arquitectura.....	24
1.3.9 Frameworks	26
1.3.10 Herramientas utilizadas en la propuesta de solución.....	28
1.4 Conclusiones	30
Capítulo 2 Características del sistema. Elaboración y Planificación	32
2.1. Introducción	32
2.2. Flujo actual de los procesos	32
2.3. Objeto de automatización	32
2.4. Propuesta del sistema	33
2.4.1. Personal relacionado con el sistema.....	34
2.5. Fase de Exploración	35
2.5.1. Historias de Usuario (HU)	35
2.6. Fase de Planificación	41
2.6.1. Estimación de esfuerzo por Historias de Usuario.....	41
2.6.2. Plan de iteraciones	42
2.6.3. Plan de duración de las iteraciones	44
2.6.4. Plan de entrega	44
2.7. Conclusiones	46
Capítulo 3 Implementación y Prueba	47
3.1. Introducción	47
3.2. Diseño del sistema	47
3.2.1. Tarjetas CRC.....	48
3.2.2. Modelo de datos	55
3.3. Implementación	56
3.3.1. Iteración 1.....	57
3.3.2. Iteración 2.....	61
3.3.3. Iteración 3.....	65
3.3.4. Iteración 4.....	67
3.4. Prueba	68

3.4.1. Pruebas de aceptación	69
3.5. Conclusiones	90
Conclusiones	91
Recomendaciones	92
Referencias Bibliográficas	93
Bibliografía	94
Glosario	96

Índice de Tablas

TABLA 1. PERSONAL RELACIONADO CON EL SISTEMA.....	34
TABLA 2. HU GESTIONAR DEPORTE	36
TABLA 3. HU GESTIONAR DELEGACIÓN.....	36
TABLA 4. HU CONFORMAR EDICIONES DE JUEGOS DEPORTIVOS	36
TABLA 5. HU GESTIONAR EQUIPO	37
TABLA 6. HU GESTIONAR COMPETENCIA	37
TABLA 7. HU GESTIONAR GRUPO.....	37
TABLA 8. HU GESTIONAR ENFRENTAMIENTO	38
TABLA 9. HU GESTIONAR EVENTO	38
TABLA 10. HU GESTIONAR RESULTADOS	38
TABLA 11. HU CONFECCIONAR TABLA DE POSICIÓN	39
TABLA 12. HU GESTIONAR RECLAMACIÓN	39
TABLA 13. HU GESTIONAR MODALIDAD	40
TABLA 14. HU GESTIONAR DIVISIÓN.....	40
TABLA 15. HU GESTIONAR USUARIO	40
TABLA 16. HU AUTENTICAR	41
TABLA 17. ESTIMACIÓN DE ESFUERZO POR HU	41
TABLA 18. PLAN DE DURACIÓN DE LAS ITERACIONES	44
TABLA 19. MÓDULOS E HU ABARCADAS	44
TABLA 20. PLAN DE DURACIÓN DE ENTREGAS.....	45
TABLA 21. TARJETA CRC CLASE COMPETENCIA.....	48
TABLA 22. TARJETA CRC CLASE DEPORTE.....	49
TABLA 23. TARJETA CRC CLASE ENFRENTAMIENTO	49
TABLA 24. TARJETA CRC CLASE EQUIPO	49
TABLA 25. TARJETA CRC CLASE EVENTO.....	50
TABLA 26. TARJETA CRC CLASE EVENTOMODALIDAD.....	50
TABLA 27. TARJETA CRC CLASE EVENTODIVISION.....	51
TABLA 28. TARJETA CRC CLASE FACULTAD	51
TABLA 29. TARJETA CRC CLASE GRUPO.....	51
TABLA 30. TARJETA CRC CLASE JUEGO.....	52
TABLA 31. TARJETA CRC CLASE JUEGOFACULTAD.....	52
TABLA 32. TARJETA CRC CLASE MODALIDAD	53
TABLA 33. TARJETA CRC CLASE RECLAMACIÓN.....	53
TABLA 34. TARJETA CRC CLASE DIVISIÓN	54
TABLA 35. TARJETA CRC CLASE USUARIO	54
TABLA 36. TARJETA CRC CLASE ATLETA.....	54
TABLA 37. TARJETA CRC CLASE ATLETAEVENTO.....	55
TABLA 38. TRANSFORMACIÓN ORM.....	55
TABLA 39. MÓDULOS ABORDADOS EN LA PRIMERA ITERACIÓN	57
TABLA 40. TAREA 1 MÓDULO DEPORTE	57
TABLA 41. TAREA 2 MÓDULO DEPORTE	57
TABLA 42. TAREA 3 MÓDULO DEPORTE	58
TABLA 43. TAREA 4 MÓDULO DEPORTE	58

TABLA 44. TAREA 1 MÓDULO DELEGACIÓN.....	58
TABLA 45. TAREA 2 MÓDULO DELEGACIÓN.....	59
TABLA 46. TAREA 3 MÓDULO DELEGACIÓN	59
TABLA 47. TAREA 3 MÓDULO DELEGACIÓN.....	59
TABLA 48. TAREA 1 MÓDULO EDICIÓN	60
TABLA 49. TAREA 2 MÓDULO EDICIÓN	60
TABLA 50. TAREA 1 MÓDULO EDICIÓN	60
TABLA 51. MÓDULOS ABORDADOS EN LA SEGUNDA ITERACIÓN	61
TABLA 52. TAREA 1 MÓDULO COMPETENCIA	61
TABLA 53. TAREA 2 MÓDULO COMPETENCIA	61
TABLA 54. TAREA 3 MÓDULO COMPETENCIA	62
TABLA 55. TAREA 4 MÓDULO COMPETENCIA	62
TABLA 56. TAREA 1 MÓDULO PROGRAMA COMPETENCIA	62
TABLA 57. TAREA 2 MÓDULO PROGRAMA COMPETENCIA	63
TABLA 58. TAREA 3 MÓDULO PROGRAMA COMPETENCIA	63
TABLA 59. TAREA 4 MÓDULO PROGRAMA COMPETENCIA	64
TABLA 60. TAREA 1 MÓDULO RESULTADO.....	64
TABLA 61. TAREA 2 MÓDULO RESULTADO.....	64
TABLA 62. TAREA 3 MÓDULO RESULTADO.....	65
TABLA 63. MÓDULOS ABORDADOS EN LA TERCERA ITERACIÓN	65
TABLA 64. TAREA 1 MÓDULO EMULACIÓN	65
TABLA 65. TAREA 2 MÓDULO EMULACIÓN	66
TABLA 66. TAREA 3 MÓDULO EMULACIÓN	66
TABLA 67. TAREA 4 MÓDULO EMULACIÓN	66
TABLA 68. MÓDULOS ABORDADOS EN LA CUARTA ITERACIÓN	67
TABLA 69. TAREA 1 MÓDULO USUARIO	67
TABLA 70. TAREA 2 MÓDULO USUARIO	67
TABLA 71. TAREA 1 MÓDULO USUARIO	68
TABLA 72. PRUEBA 1 DEL MÓDULO DEPORTE	69
TABLA 73. PRUEBA 2 DEL MÓDULO DEPORTE	69
TABLA 74. PRUEBA 3 DEL MÓDULO DEPORTE	70
TABLA 75. PRUEBA 1 DEL MÓDULO DELEGACIÓN	70
TABLA 76. PRUEBA 2 DEL MÓDULO DELEGACIÓN	71
TABLA 77. PRUEBA 3 DEL MÓDULO DELEGACIÓN	71
TABLA 78. PRUEBA 4 DEL MÓDULO DELEGACIÓN	72
TABLA 79. PRUEBA 5 DEL MÓDULO DELEGACIÓN	73
TABLA 80. PRUEBA 6 DEL MÓDULO DELEGACIÓN	73
TABLA 81. PRUEBA 7 DEL MÓDULO DELEGACIÓN	74
TABLA 82. PRUEBA 8 DEL MÓDULO DELEGACIÓN	74
TABLA 83. PRUEBA 1 DEL MÓDULO EDICIÓN	75
TABLA 84. PRUEBA 2 DEL MÓDULO EDICIÓN	75
TABLA 85. PRUEBA 1 DEL MÓDULO COMPETENCIA	76
TABLA 86. PRUEBA 2 DEL MÓDULO COMPETENCIA	76
TABLA 87. PRUEBA 3 DEL MÓDULO COMPETENCIA	77
TABLA 88. PRUEBA 4 DEL MÓDULO COMPETENCIA	78
TABLA 89. PRUEBA 5 DEL MÓDULO COMPETENCIA	78
TABLA 90. PRUEBA 6 DEL MÓDULO COMPETENCIA	79

TABLA 91. PRUEBA 7 DEL MÓDULO COMPETENCIA	79
TABLA 92. PRUEBA 8 DEL MÓDULO COMPETENCIA	80
TABLA 93. PRUEBA 9 DEL MÓDULO COMPETENCIA	80
TABLA 94. PRUEBA 1 DEL MÓDULO PROGRAMA COMPETENCIA	81
TABLA 95. PRUEBA 2 DEL MÓDULO PROGRAMA COMPETENCIA	82
TABLA 96. PRUEBA 3 DEL MÓDULO PROGRAMA COMPETENCIA	82
TABLA 97. PRUEBA 4 DEL MÓDULO PROGRAMA COMPETENCIA	83
TABLA 98. PRUEBA 5 DEL MÓDULO PROGRAMA COMPETENCIA	83
TABLA 99. PRUEBA 6 DEL MÓDULO PROGRAMA COMPETENCIA	84
TABLA 100. PRUEBA 7 DEL MÓDULO PROGRAMA COMPETENCIA.....	85
TABLA 101. PRUEBA 8 DEL MÓDULO PROGRAMA COMPETENCIA.....	85
TABLA 102. PRUEBA 9 DEL MÓDULO PROGRAMA COMPETENCIA.....	86
TABLA 103. PRUEBA 1 DEL MÓDULO RESULTADO	86
TABLA 104. PRUEBA 2 DEL MÓDULO RESULTADO	87
TABLA 105. PRUEBA 3 DEL MÓDULO RESULTADO	88
TABLA 106. PRUEBA 3 DEL MÓDULO RESULTADO	88
TABLA 107. PRUEBA 1 DEL MÓDULO EMULACIÓN.....	89
TABLA 108. PRUEBA 2 DEL MÓDULO EMULACIÓN.....	89

Índice de Figuras

FIGURA 1 METODOLOGÍA RUP	21
FIGURA 2 METODOLOGÍA XP	22
FIGURA 3 ARQUITECTURA DE 3 CAPAS.....	24
FIGURA 4 PATRÓN MVC	25
FIGURA 5. MODELO DE DATOS.....	56

Introducción

En el mes de marzo del año 2002 las primeras ideas del “Comandante en Jefe Fidel Castro Ruz” acerca de la creación de una “Universidad de Nuevo Tipo”, de una “Universidad de Excelencia”, comenzaron a cristalizar a través del denominado “Proyecto Futuro”. Habían transcurrido apenas siete meses, cuando el 23 de septiembre del propio año el sueño se hizo realidad, pues la Universidad de las Ciencias Informáticas (UCI), abrió sus puertas a los primeros 2008 jóvenes provenientes de todo el país.

Desde su surgimiento, la UCI, en su carácter de universidad científica, tecnológica y humanística, hizo evidente la formación integral de los futuros egresados, capaces de ofrecer una mejor respuesta a las necesidades económicas y sociales del país, por lo que para el logro de sus procesos de formación, investigación y extensión en integración dialéctica, se proporcionaron los espacios necesarios como garantía del cumplimiento de la misión de las universidades en nuestro país.

En su devenir y como parte de los procesos de extensión universitaria, se potenció también desde los inicios la práctica del deporte y el desarrollo de espacios para la recreación sana de los estudiantes, contribuyendo así a elevar su calidad de vida. Actividad que en el transcurso de los años se ha ido perfeccionando e incrementado sus resultados a niveles tales, que hoy ya se participa en calidad de invitados en la Olimpiada del Deporte Cubano en algunos eventos como el fútbol sala y el kárate-do.

Uno de los eventos más emocionantes de la vida universitaria y que forma parte de las actividades de extensión universitaria, lo constituyen los Juegos Deportivos Interfacultades, espacio de competición fraternal donde cada facultad defiende con pasión los colores que la identifican y para el cual se preparan los atletas desde los inicios del curso. Dada la aceptación del evento, avalado por el poder movilizativo que posee para participar, ya sea en calidad de espectador o como atleta, la organización de los mismos se va haciendo cada vez más compleja, agudizado por los incrementos de matrícula y la incorporación de nuevas disciplinas al calendario deportivo. Esta situación ha llevado a la idea de desarrollar una aplicación Web para los Juegos Interfacultades con el objetivo de facilitar la gestión de los procesos de información inherentes a dichos juegos.

En los momentos actuales, horas antes del comienzo del evento, se “monta” un Portal cuya funcionalidad no cubre las necesidades reales identificadas, motivo por el cual persisten numerosos problemas que atentan contra la eficiente gestión de los juegos deportivos. Entre las dificultades

detectadas se pueden citar que los procesos no se automatizan del todo por lo cual el trabajo continúa siendo engorroso, repetitivo y muchas veces realizado de manera manual. Se necesita de un estudiante con conocimientos informáticos que se encargue del funcionamiento del sistema debido a que el mismo no ofrece facilidades para gestionar de manera dinámica muchos de los procesos, lo cual obliga a los usuarios a realizar cambios directos en el editor de texto e incluso en la base de datos. La implementación del sistema además, se basa en el uso del CMS Drupal el cual no ha sido diseñado con módulos específicos para la gestión de juegos deportivos. Otro de los problemas lo constituye el que, una vez concluido el evento, los datos relacionados al mismo se pierden de manera que su consulta en años posteriores resulta imposible.

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, nos proponemos el siguiente **problema científico**:

- ¿Cómo facilitar de forma eficiente, segura y rápida la gestión de los procesos de información de los Juegos Deportivos Interfacultades de la Universidad de las Ciencias Informáticas?

El **objeto de estudio** de esta investigación es:

- Sistemas para la gestión de juegos deportivos.

Dicho objeto enmarca el **campo de acción** de esta investigación:

- Gestión de los procesos de información de los Juegos Deportivos Interfacultades en la Universidad de las Ciencias Informáticas.

El **objetivo** principal de esta investigación es:

- Desarrollar un sistema informático que gestione los procesos relacionados con la información de los Juegos Deportivos Interfacultades de la Universidad de las Ciencias Informáticas.

Los **objetivos específicos** que se persiguen son:

- Procesar toda la información inherente a la gestión de los Juegos Deportivos Interfacultades.
- Realizar diseño, implementación y prueba de un sistema que gestione los procesos relacionados con la información de los Juegos Deportivos Interfacultades.
- Crear un documento que recoja todo el proceso de desarrollo del sistema informático.

Ideas a defender

Si se desarrolla un sistema que gestione los procesos de información inherentes a los Juegos Deportivos Interfacultades en la Universidad de Ciencias Informáticas la organización, control, procesamiento y actualización de los mismos se realizarán de manera más rápida, segura, efectiva y organizada logrando así su optimización.

Tareas a cumplir para lograr los objetivos propuestos:

1. Realizar un estudio sobre la existencia de aplicaciones o soluciones similares con las características del sistema propuesto.
2. Realizar un estudio sobre las tendencias y tecnologías que actualmente se utilizan en la producción de software.
3. Definir la metodología que guiará el proceso de desarrollo del software.
4. Seleccionar las tecnologías y herramientas para la implementación del software teniendo en cuenta las características del mismo.
5. Recopilar la información relacionada con la Gestión de los Procesos de los Juegos Deportivos Interfacultades para establecer el funcionamiento del sistema a implementar.
6. Realizar la planificación para la posterior implementación del sistema.
7. Implementar el sistema "Sistema para la Gestión de la Información de los Juegos Deportivos Interfacultades".
8. Realizar pruebas al sistema.
9. Estructurar la investigación realizada en el formato requerido para obtener una buena documentación.

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. Por otra parte, 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, centrando la atención en las historias de usuario. Además se construyen otros 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

Hace tan solo unos años imaginar el increíble desarrollo alcanzado a nivel mundial en lo referente a las ciencias informáticas hubiera sido un sueño. El hombre, emprendedor por naturaleza, ha hecho transitar este sueño hasta convertirlo en una realidad que no deja de asombrarnos día a día con nuevos avances. Dentro del mundo de la informática, específicamente en la producción de software, es común observar atónitos como lo que hasta ayer parecía la mejor solución hoy da paso a otras que nacen sin imaginar que tal vez mañana serán a su vez reemplazadas. 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”.

El siguiente capítulo es el resultado de una detallada y profunda investigación acerca de los conceptos esenciales asociados al entorno del problema de la presente investigación, de la actualidad que posee la misma, así como de 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 mundo actual vive inmerso en la “Era de la Información” y en consecuencia, las organizaciones en su desarrollo han aprendido a gestionar la misma. La información es un elemento fundamental para el desarrollo sin embargo uno de los principales problemas en la actualidad es su exceso, de manera que filtrar la información realmente relevante se torna complicado. Es aquí donde la **gestión de la información** se hace imprescindible ocupando un espacio mayor en la estrategia de los países a escala mundial. (1)

A partir de lo anterior se concluye que la **gestión de la información** comprende las actividades relacionadas con la obtención de la información adecuada, a un precio adecuado, en el tiempo y lugar adecuado, para tomar la decisión adecuada.

Deporte

La palabra deporte evidencia un fenómeno sociocultural y educativo por todos conocidos, sin embargo muchas definiciones se han escrito sobre la misma, convirtiéndola en un término tan fascinante como complejo.

En Cuba, a partir del triunfo de la Revolución el deporte deja de ser privilegio de unos pocos para convertirse en práctica cotidiana de muchos, por lo que sustentados en la idea de que “El deporte, es derecho del pueblo” ha alcanzado gran masividad y su práctica se extiende, a escuelas y áreas deportivas especiales donde se propicia el desarrollo de capacidades y habilidades motrices en los estudiantes.

El **deporte** es toda aquella actividad que se caracteriza por: tener un requerimiento físico o motriz, estar institucionalizado (federaciones, clubes), requerir competición con uno mismo o con los demás y tener un conjunto de reglas perfectamente definidas.

El Barón Pierre de Coubertin, “Padre del Olimpismo Moderno”, en sus “Memorias Olímpicas” declara: “El deporte no es un artículo de lujo, no es una ocupación para ociosos ni una compensación por el trabajo intelectual. El deporte es una fuente de perfeccionamiento interno para cada persona. La profesión no tiene nada que ver con ello. Antes bien, el deporte es un regalo irremplazable que le es dado a todas las personas en igual medida. Desde una perspectiva étnica tampoco existe diferencia, ya que, por naturaleza, todas las razas disponen del deporte como de algo propio y en igualdad de derecho”. (2)

Juegos Deportivos Interfacultades

Es el evento deportivo más grande que se realiza anualmente dentro de la UCI, en ellos participan todas las Facultades de la Universidad y a partir del año 2008 ya se contó con una representación de sus Facultades Regionales en Artemisa, Ciego de Ávila y Granma. Los Juegos Interfacultades tienen como fin, agrupar lo mejor del talento deportivo interno, para conformar los diferentes equipos por disciplinas deportivas, con vistas a la participación en los Juegos Provinciales Universitarios, además de promover 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.

1.2.2 Sistemas informáticos similares

Desde hace ya varios años, 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 se ha ido progresivamente 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**¹, 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**² y **Fox**³.

En Cuba, el volumen de sistemas informáticos relacionados a la gestión de información deportiva no alcanza los niveles internacionales. Entre los existentes podemos destacar el sistema **Estudio de Contrarios de Beisbol**, 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 **béisbol.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. Por otra parte no se tiene conocimiento de la existencia de un sistema informático, de producción nacional, encargado de gestionar la información relacionada a eventos deportivos múltiples, he aquí la necesidad del presente trabajo.

¹ Sitio oficial de la ESPN, <http://espn.go.com/>

² Sitio de deporte de Yahoo , <http://sports.yahoo.com/>

³ Sitio de deporte de la cadena Fox, <http://msn.foxsports.com/>

1.3 Tendencias y tecnologías actuales

1.3.1 Política de migración al Software Libre

El software libre es aquel que puede ser distribuido, modificado, copiado y usado; por lo tanto, debe venir acompañado del código fuente para hacer efectivas las libertades que lo caracterizan. Es necesario no confundir el software libre con el software gratuito. El hecho de que este último no cueste nada no lo convierte en libre porque no es una cuestión de precio sino de libertad. Las cuatro libertades que proponen, las cuales constituyen un derecho no una obligación, son:

- La libertad para ejecutar el programa sea cual sea nuestro propósito.
- La libertad para estudiar el funcionamiento del programa y adaptarlo a tus necesidades (el acceso al código fuente es condición indispensable para esto).
- La libertad para redistribuir copias y ayudar así a tu vecino.
- La libertad para mejorar el programa y luego publicarlo para el bien de toda la comunidad (el acceso al código fuente es condición indispensable para esto).

El software libre ha dejado de ser una promesa para convertirse en una realidad. Hoy en día muchas de las más importantes empresas tecnológicas como IBM, SUN Microsystems, Google y Hewlett-Packard lo utilizan en sistemas de producción e incluso, aunque suene contradictorio, la propia Microsoft, poseedora de sus herramientas, emplea GNU Linux en muchos de sus servidores. Por tal motivo el futuro del software libre se vislumbra consolidado y seguro.

Sin lugar a dudas el software libre constituye una alternativa para nuestro país atendiendo a dos aspectos fundamentalmente:

Político: El software libre es por concepción la alternativa de los países pobres basado en sus políticas de propiedad social y desarrollo cooperativo así como en su objetivo de beneficiar a toda la comunidad.

Económico: Su utilización no implica gastos adicionales por concepto de cambio de plataforma de software, por cuanto es operable en el mismo soporte de hardware con que cuenta el país.

1.3.2 Tipos de aplicaciones

En la actualidad con el desarrollo de las tecnologías y el aumento de las computadoras, existe una creciente tendencia al uso de herramientas informáticas, tales como las aplicaciones Web y las de Escritorio, en aras de lograr soluciones optimizadas a diversos problemas. A continuación se justifica la selección del tipo de aplicación que más se adapta a los requerimientos del problema de la presente investigación.

Aplicaciones Web vs Aplicaciones Escritorio

Se denomina aplicación Web a aquellas aplicaciones a las cuales acceden los usuarios a través de Internet o Intranet mediante un navegador. (3)En lo que respecta a su arquitectura, se distinguen dos lados, el 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 distribuir las 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 AJAX⁴. 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.

⁴ Asynchronous JavaScript And XML

- 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. (4)

1.3.3 Lenguajes de programación Web

Con el auge de Internet los lenguajes de programación para la Web se han diversificado, motivo por el cual a la hora de elegir el lenguaje que se utilizará 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 (eXtensible HyperText Markup Language)

El lenguaje XHTML es muy similar al lenguaje HTML, pues constituye una versión más estricta y limpia del mismo, adaptándolo al lenguaje XML. (5) 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. (6)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. (7) 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 “PHP Hypertext Pre-processor” 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. (8)

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. (9)

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

PHP fue diseñado desde cero con el objetivo de desarrollar aplicaciones Web. Esto quiere decir que las tareas más habituales para la realización de estas 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.

Entre las desventajas que más se le atribuyen al lenguaje PHP resaltan:

- Por sus características promueve la creación de código desordenado y complejo de mantener.
- Está diseñado especialmente para un modo de hacer aplicaciones Web que es ampliamente considerado problemático y obsoleto (mezclar el código con la creación de la página Web).

Sin embargo, las características antes planteadas son del todo subjetivas, pues dependen en gran medida de los desarrolladores; si los mismos son disciplinados y se preocupan por diseñar previamente lo que quieren antes de comenzar a teclear, las mismas no tienen sentido. Es una realidad que 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. En estos momentos el uso de patrones y el desarrollo de frameworks para PHP, aspectos que se verán más adelante, garantizan una programación sumamente ordenada, sencilla y lo que es mejor aún, divertida.

Por tanto, se selecciona el lenguaje **PHP** como Lenguaje de Programación del lado del Servidor atendiendo, además, a una serie de características positivas respecto a otros lenguajes.

- Es un lenguaje multiplataforma.
- Rapidez de ejecución.
- Mantiene un bajo consumo de recursos de máquina.
- Gran seguridad, muy poca probabilidad de corromper los datos.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Posee una amplia documentación en Internet, incluyendo una gran variedad de ejemplos y de ayudas.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- No requiere definición de tipos de variables.

1.3.4 AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo Web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas se ejecutan en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

AJAX no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente:

- XHTML(o HTML) y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano.

Por tanto, se selecciona **AJAX** para su uso como parte de la propuesta de solución atendiendo, además, a una serie de características positivas.

- Recuperación asíncrona de datos, el usuario no tiene que esperar después de una petición.
- Acercamiento del ambiente de escritorio a la Web.
- No requiere plug-ins(es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica).
- Se reduce el tamaño de la información intercambiada.

1.3.5 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. Aún 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. (10)

PostgreSQL

Está considerado el SGBD de software libre más avanzado del mundo, publicado bajo la licencia BSD⁵. 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.

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

⁵ Berkeley Software Distribution, licencia de software perteneciente al grupo de licencias de software libre.

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, además, a una serie de características positivas respecto a otros gestores.

- 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.

1.3.6 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 solemos conocer 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.

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 atendiendo, además, a una serie de características positivas respecto a otros servidores.

- Modular
- Multiplataforma
- Software Libre
- Extensible
- Popular
- Gratuito

1.3.7 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 acreditan. 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 (puesta en marcha)
- Elaboración(definición, análisis y diseño)
- Construcción(implementación)
- Transición (fin del proyecto y puesta en producción)

El ciclo de vida de RUP se caracteriza por:

1. **Dirigido por casos de uso:** 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).
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.
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.

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:** Trasladando las necesidades del negocio a un sistema automatizado.
- **Análisis y Diseño:** Trasladando los requerimientos dentro de la arquitectura de software.
- **Implementación:** Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- **Pruebas:** Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Flujos de apoyo:

- **Configuración y administración del cambio:** Guardando todas las versiones del proyecto.
- **Administrando el proyecto:** Administrando horarios y recursos.
- **Ambiente:** Administrando el ambiente de desarrollo.

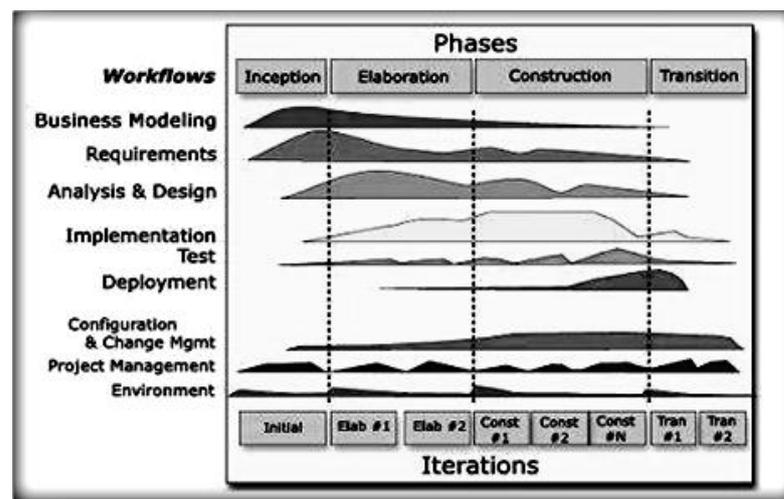


Figura 1 Metodología RUP

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierta luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendrá en cada entregable o en cada iteración.

XP (eXtreme Programing)

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 claves 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 añada 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. (12)

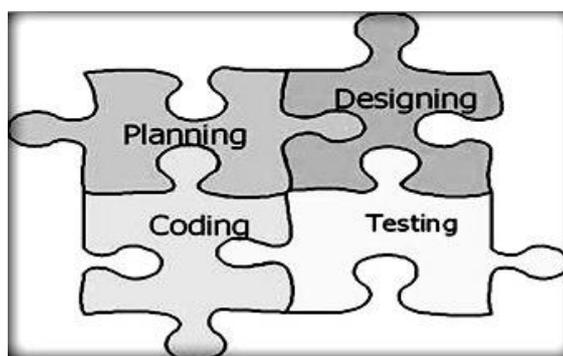


Figura 2 Metodología XP

Selección de la Metodología de Software

Como se ha analizado RUP pertenece al grupo de las metodologías robustas y como tal basa su éxito en mantener una documentación ordenada a lo largo de todo el proceso de desarrollo. Por su parte XP, 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 los clientes.

Por tanto, se selecciona **XP** como metodología de desarrollo de software atendiendo, además, a que se adapta en gran medida, tanto al tipo de proyecto a desarrollar como a las condiciones de trabajo.

- El proyecto es pequeño: XP está concebida para ser utilizada dentro de proyectos pequeños.
- Los requisitos del cliente 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.
- El riesgo de desarrollo es elevado debido al corto tiempo de entrega planteado y a los continuos cambios de requerimientos: XP está diseñada a mitigar los riesgos en proyectos con estas características.
- 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. Enfatiza la comunicación de los programadores a través del código, utilizando líneas directivas para la codificación que están bien establecidas.

1.3.8 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 en 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.

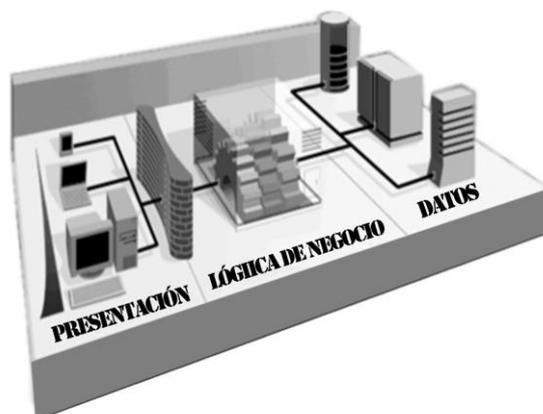


Figura 3 Arquitectura de 3 capas

Modelo-Vista-Controlador (MVC)

Un propósito común en numerosos sistemas es el de tomar datos almacenados y mostrarlos al usuario. Luego que el usuario introduce modificaciones, las mismas se reflejan en el almacenamiento. Dado que el flujo de información ocurre entre el almacenamiento y la interfaz, una tentación común, un impulso espontáneo es unir ambas piezas para reducir la cantidad de código y optimizar el rendimiento.

Sin embargo, esta idea no es correcta por el hecho de que la interfaz suele cambiar, o acostumbra depender de distintas clases de dispositivos (aplicaciones de ventana, navegadores, dispositivos inalámbricos); la programación de interfaces de XHTML requiere habilidades muy distintas de la

programación de lógica de negocios. Otro problema es que las aplicaciones tienden a incorporar lógica de negocios que van más allá de la transmisión de datos.

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

- El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La Vista transforma el modelo en una página Web que permite al usuario interactuar con ella.
- El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

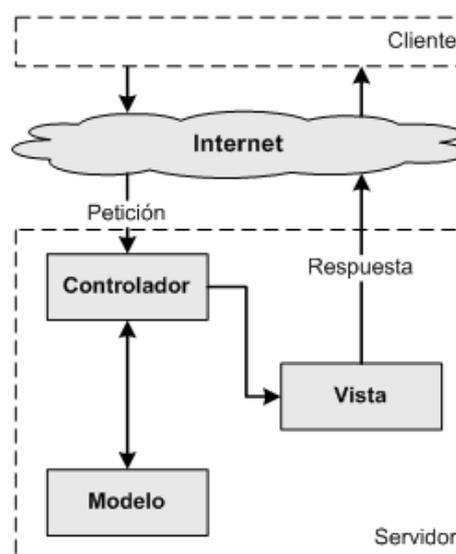


Figura 4 Patrón MVC

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo, independientemente de la representación visual.

Selección de la Arquitectura

Como propuesta de solución se plantea el uso de la arquitectura Modelo-Vista-Controlador atendiendo, además, a las ventajas indudables que proporciona.

- Soporte de múltiples vistas: Dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente. Por ejemplo, múltiples páginas de una aplicación Web pueden utilizar el mismo modelo de objetos mostrado de maneras diferentes.

- Adaptación al cambio: Los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Los usuarios pueden preferir distintas opciones de representación, o requerir soporte para nuevos dispositivos como teléfonos celulares. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.

1.3.9 Frameworks

Un framework 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. (13)

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.

symfony⁶

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. (13)

⁶ A partir de la versión 2.0 se nombrará Symfony con 'S' mayúscula, la versión actual es la 1.2.4

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.
- Gran cantidad de tutoriales y ejemplos de código disponible tanto en Internet como en la UCI.

1.3.10 Herramientas 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. (14)

Eclipse

Es un IDE de programación totalmente 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 auto-completamiento y colorea errores. Actualmente se encuentra disponible la versión 2.0 de los PHP Development Tools (PDT) de Eclipse, los componentes especialmente diseñados para facilitar el desarrollo en PHP de acuerdo a 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 2.0 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). (15)

EasyEclipse

Este es un IDE que facilita el desarrollo de aplicaciones Web, su objetivo principal es la facilidad de instalación. (16)

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. (17)

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

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. Aún 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 con la incorporación del nuevo plug-in para PHP denominado PDT 2.0 y a la familiarización que se posee con la misma.

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.

Capítulo 2 Características del sistema. Elaboración y Planificación

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 en aras de lograr 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 a 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 Interfacultades 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 Interfacultades 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 tanto los procesos vinculados a la gestión del programa de competencia de los Juegos Deportivos Interfacultades, como la incorporación de los resultados a 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.

Otros de los procesos que serán objeto de automatización serán la conformación de la tabla de medallas y la de 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, las cuales se abundarán a continuación.

Dicho sistema debe ser capaz de gestionar toda la información derivada de la creación de nuevas ediciones de Juegos Deportivos Interfacultades. Por tanto debe permitir la gestión de las diferentes competencias que tienen lugar en los mismos. Las competencias serán agrupadas, atendiendo a sus características, según lo especificado a continuación:

- Competencias por grupo
- Competencias por modalidad
- Competencias por división
- Otras competencias

En el caso de las competencias por grupo se brindará la posibilidad de organizar las facultades participantes en diferentes grupos para que los mismos estén disponibles a la hora de gestionar los enfrentamientos.

Para el caso de las competencias por modalidad el sistema deberá permitir gestionar los diferentes eventos por modalidad que se desarrollen dentro del marco de este tipo de competencia.

Las competencias por división serán similares a las de modalidad con la particularidad de que de ellas se controlarán las diferentes divisiones en las cuales se compiten.

Por último el sistema debe brindar la posibilidad de gestionar otro tipo de competencias, estas son las que no presentan un sistema competitivo definido, por cuanto solo interesa controlar los resultados finales de las mismas.

Independientemente del tipo de competencia la aplicación en cuestión deberá facilitar la entrada de resultados, éstos serán dados tanto de forma individual como colectiva, según las particularidades de cada competencia. Además el sistema debe dar la posibilidad de agregar y quitar tanto medallas como puntos a las diferentes facultades para que sea posible establecer las posiciones finales en la tabla de medallas y en la de emulación.

Otras de las funcionalidades generales con las cuales el sistema deberá contar están vinculadas a la gestión de las delegaciones que participarán en los juegos, incorporándose por defecto las 10 facultades existentes. También debe ser capaz de gestionar la información de las diferentes disciplinas deportivas en las que se competirá así como de los equipos conformados por cada facultad para incursionar en las mismas.

Con el fin de definir los permisos de cada usuario se debe brindar la posibilidad de gestionar los mismos recogiendo sus datos de interés y asignándosele un rol dentro de la aplicación.

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”, a menudo la más importante, tendrá como objetivo fundamental permitir la gestión de la misma. A partir del análisis de las características que el sistema en cuestión debe poseer el mismo implementará la aplicación “trasera” 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, a todo usuario que obtiene un resultado de valor de al menos uno de los procesos que se ejecutan en el mismo.

A partir de la propuesta de diseño elaborada se hace necesario la presencia de tres grupos de usuarios: Comisión organizadora, Comisión de emulación y Administrador.

Tabla 1. Personal relacionado con el sistema

Personal relacionado con el sistema	Descripción
Comisión organizadora	Este es el personal encargado de realizar todas las tareas de organización referente a los Juegos Deportivos

	Interfacultades. Está autorizado a realizar todos los cambios necesarios dentro del proceso de los mismos, ya sea inscribir las delegaciones, equipos y deportes. Además le es permitido adicionar una nueva edición de los juegos así como editarla. Por otra parte es el encargado de crear el cronograma de competencias, confeccionar la tabla de medallas e introducir resultados.
Comisión de emulación	Este es el usuario que va a tener solo los privilegios de acceder a confeccionar tanto la tabla de medallas como la tabla de puntuación, además de le es permitida la gestión de reclamaciones.
Administrador	Es la persona facultada para la gestión del sistema en general. Es el encargado de asignar y administrar las diferentes cuentas de los usuarios autenticados en la aplicación. Además cuenta con los permisos que le son dados al personal Comisión organizadora y Usuario anónimo.

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. (18)

Tabla 2. HU Gestionar deporte

Historia de Usuario	
No. : 1	Nombre: Gestionar deporte.
Usuario : Administrador, Comisión organizadora	
Prioridad en el Negocio: Alta	Riesgo de desarrollo : Bajo
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: Se realiza la acción de crear, modificar y eliminar un deporte. El sistema debe dar la posibilidad de añadir un deporte que no se encuentre registrado. Además de permitir tanto modificar como eliminar un deporte existente.	
Observación:	

Tabla 3. HU Gestionar delegación

Historia de Usuario	
No. : 2	Nombre: Gestionar delegación.
Usuario : Administrador, Comisión organizadora	
Prioridad en el Negocio: Alta	Riesgo de desarrollo : Bajo
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: Se realiza la acción de crear, modificar o eliminar una delegación. El sistema debe brindar la posibilidad de añadir una delegación así como de modificar o eliminar dicha delegación.	
Observación: Por defecto las 10 facultades existentes aparecerán inscritas.	

Tabla 4. HU Conformar ediciones de Juegos Deportivos

Historia de Usuario	
No. : 3	Nombre: Conformar ediciones de Juegos Deportivos.
Usuario : Administrador, Comisión organizadora	
Prioridad en el Negocio: Alta	Riesgo de desarrollo : Bajo
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: Se realiza la acción de crear y modificar ediciones de juegos deportivos. El sistema debe dar la posibilidad de crear una nueva edición de los Juegos Deportivos Interfacultades así como modificar la misma.	
Observación: Las delegaciones que participarán deben estar inscritas previamente.	

Tabla 5. HU Gestionar equipo

Historia de Usuario	
No. : 4	Nombre: Gestionar equipo.
Usuario : Administrador, Comisión organizadora	
Prioridad en el Negocio: Alta	Riesgo de desarrollo : Medio
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: Se realizan las acciones de crear, modificar y eliminar un equipo. El sistema debe brindar la posibilidad de conformar un equipo perteneciente a deporte determinado dada la delegación. Además debe dar la opción de modificar los datos referentes a un equipo existente así como eliminarlo.	
Observación:	

Tabla 6. HU Gestionar competencia

Historia de Usuario	
No. : 5	Nombre: Gestionar competencia.
Usuario : Administrador, Comisión organizadora	
Prioridad en el Negocio: Alta	Riesgo de desarrollo : Alto
Puntos de Estimación: 1	Iteración Asignada: 2
Descripción: Se realizan las acciones de crear, modificar y eliminar una competencia. El sistema debe brindar la opción de crear una competencia de cualquier deporte dándole la posibilidad de seleccionar el tipo de competencia que desea confeccionar. Además debe dar la opción de modificar o eliminar los datos relacionados con competencias existentes.	
Observación:	

Tabla 7. HU Gestionar grupo

Historia de Usuario	
No. : 6	Nombre: Gestionar grupos.
Usuario : Administrador, Comisión organizadora	
Prioridad en el Negocio: Alta	Riesgo de desarrollo : Medio
Puntos de Estimación: 1	Iteración Asignada: 2
Descripción: Se realizan las acciones de crear, modificar y eliminar un grupo. El sistema debe brindar al usuario la opción de conformar los grupos necesarios para una competencia	

determinada. Además debe dar la opción de modificar dichos grupos así como eliminarlos.
Observación: Previamente debe estar creada alguna competencia de grupo.

Tabla 8. HU Gestionar enfrentamiento

Historia de Usuario	
No. : 7	Nombre: Gestionar enfrentamiento.
Usuario : Administrador, Comisión organizadora	
Prioridad en el Negocio: Alta	Riesgo de desarrollo : Medio
Puntos de Estimación: 1	Iteración Asignada: 2
Descripción: Se realizan las acciones de crear, modificar y eliminar enfrentamientos. Se debe brindar la posibilidad al usuario de crear un enfrentamiento entre dos equipos participantes en una competencia determinada. Además se debe dar la opción de modificar dicho enfrentamiento así como eliminarlo.	
Observación: Solo es válido para las competencias por grupo.	

Tabla 9. HU Gestionar evento

Historia de Usuario	
No. : 8	Nombre: Gestionar evento.
Usuario : Administrador, Comisión organizadora	
Prioridad en el Negocio: Alta	Riesgo de desarrollo : Medio
Puntos de Estimación: 1	Iteración Asignada: 2
Descripción: Se realizan las acciones de crear, modificar y eliminar un evento. El sistema debe brindar la posibilidad al usuario de crear un evento de una modalidad o división determinada, según el deporte al que pertenezca la competencia. Además se debe dar la posibilidad al usuario de modificar o eliminar los datos de dicho evento.	
Observación: Solo es válido para las competencias por modalidad o división.	

Tabla 10. HU Gestionar resultados

Historia de Usuario	
No. : 9	Nombre: Gestionar resultados
Usuario : Administrador, Comisión organizadora	

Prioridad en el Negocio: Alta	Riesgo de desarrollo :Alto
Puntos de Estimación: 2	Iteración Asignada: 2
Descripción: Se realiza las acciones de introducir, modificar y “limpiar” resultados de una competencia. El sistema debe brindar la posibilidad de introducir el resultado de una competencia seleccionada, este va a estar determinado por el tipo de competencia de la cual se quiera introducir resultados. Además debe dar la posibilidad tanto de modificar como de “limpiar” algún resultado de una competencia determinada.	
Observación:	

Tabla 11. HU Confeccionar tabla de posición

Historia de Usuario	
No. : 10	Nombre: Confeccionar tabla de posición.
Usuario : Administrador, Comisión organizadora	
Prioridad en el Negocio: Media	Riesgo de desarrollo : Medio
Puntos de Estimación: 1	Iteración Asignada: 3
Descripción: Se realizan las actividades de añadir y restar. El sistema debe brindar la posibilidad de añadir tanto medallas como puntos a una delegación determinada, así como quitar medallas y puntos a dicha delegación.	
Observación:	

Tabla 12. HU Gestionar reclamación

Historia de Usuario	
No. : 11	Nombre: Gestionar reclamación.
Usuario : Comisión de emulación	
Prioridad en el Negocio: Alta	Riesgo de desarrollo : Medio
Puntos de Estimación: 1	Iteración Asignada: 3
Descripción: Se realizan las actividades de añadir, modificar y eliminar. El sistema debe brindar la posibilidad de añadir una reclamación a la presente edición de los juegos deportivos. De igual modo se permitirá modificar o eliminar una reclamación existente.	
Observación:	

Tabla 13. HU Gestionar modalidad

Historia de Usuario	
No. : 12	Nombre: Gestionar modalidad
Usuario : Administrador, Comisión organizadora	
Prioridad en el Negocio: Alta	Riesgo de desarrollo : Bajo
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: Se realiza la acción de crear, modificar y eliminar una modalidad perteneciente a un deporte. El sistema debe dar la posibilidad de añadir una modalidad a un deporte que se encuentre registrado como deporte de modalidad. Además de permitir modificar y eliminar dicha modalidad.	
Observación: El deporte debe estar previamente inscrito.	

Tabla 14. HU Gestionar división

Historia de Usuario	
No. : 13	Nombre: Gestionar división
Usuario : Administrador, Comisión organizadora	
Prioridad en el Negocio: Alta	Riesgo de desarrollo : Bajo
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: Se realiza la acción de crear, modificar y eliminar una división perteneciente a un deporte. El sistema debe dar la posibilidad de añadir una división a un deporte que se encuentre registrado como deporte de división. Además de permitir modificar y eliminar dicha división.	
Observación:	

Tabla 15. HU Gestionar usuario

Historia de Usuario	
No. : 14	Nombre: Gestionar usuario.
Usuario : Administrador	
Prioridad en el Negocio: Alta	Riesgo de desarrollo : Alto
Puntos de Estimación: 1	Iteración Asignada: 4
Descripción: Se realiza la acción de crear, modificar o eliminar una cuenta del sistema. El sistema deberá brindar la opción de crear una cuenta y darle los permisos correspondientes a	

las competencias que tendrá el mismo dentro de la aplicación. Además debe dar la posibilidad de modificar o eliminar una cuenta de usuario seleccionada.
Observación:

Tabla 16. HU Autenticar

Historia de Usuario	
No. : 15	Nombre: Autenticar.
Usuario : Administrador, Comisión organizadora, Comisión de emulación	
Prioridad en el Negocio: Alta	Riesgo de desarrollo : Medio
Puntos de Estimación: 1	Iteración Asignada: 4
Descripción: La aplicación debe brindar la opción a cualquier usuario que acceda al sistema de introducir sus datos (usuario y contraseña) para así verificar y asignarle los permisos que le competen en la aplicación.	
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 17. Estimación de esfuerzo por HU

Historias de usuario	Puntos de estimación
Gestionar deporte	1
Gestionar delegación	1
Conformar edición de Juegos Deportivos	1
Gestionar equipo	1
Gestionar competencia	1

Gestionar grupos	1
Gestionar enfrentamientos	1
Gestionar eventos	1
Gestionar resultados	2
Confeccionar tabla de posición	1
Gestionar reclamación	1
Gestionar modalidad	1
Gestionar división	1
Gestionar usuario	1
Autenticar	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 cuatro iteraciones, las cuales se describen a continuación:

Iteración 1

Esta iteración tiene como objetivo darle cumplimiento a las historias de usuario 1, 2, 3, 4, 12 y 13. Dichas historias de usuario son de vital importancia para la aplicación, mediante ellas se va a conformar la estructura básica que va a tener el sistema, además, a partir de ellas se realizarán las demás funcionalidades. Las historias de usuario, antes mencionadas, hacen alusión, por una parte, a la gestión de ediciones de los Juegos Deportivos Interfacultades y de las delegaciones que van a participar en los mismos. Por otro lado se garantiza que existan deportes disponibles, incluidas sus modalidades o divisiones en caso de tenerlas, para crear competencias en los juegos así como gestionar toda la información relacionada con los equipos que van a participar en ellos. 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 5, 6, 7, 8, 9. La implementación de dichas historias de usuario brindará funcionalidades fundamentales del sistema. La HU antes mencionadas se relacionan con los procesos de gestionar tanto una competencia de un deporte determinado como los programas de competencia inherentes a las mismas. De igual manera se relaciona, con las HU seleccionadas para la presente iteración, la gestión de los resultados de cada una de las competencias. La entrega que se realice, una vez culminada esta iteración, en unión con la realizada en la iteración anterior, una vez más servirá para verificar con el cliente si se van cumpliendo sus expectativas.

Iteración 3

Esta iteración tiene como objetivo darle cumplimiento a las historias de usuario 10 y 11. 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: confeccionar la tabla de posición de la edición en curso de los juegos deportivos tanto en su variante de medallero como en la de puntuación y gestionar las reclamaciones asociadas a la presente edición de juegos deportivos. Al término de esta iteración se realizará una nueva entrega al cliente, la cual, en unión con las anteriores evidenciará la primera aproximación de la aplicación. En consecuencia con lo realizado en las iteraciones anteriores la retroalimentación con el cliente jugará un papel fundamental para el equipo de desarrollo.

Iteración 4

Esta iteración tiene como objetivo darle cumplimiento a las historias de usuario 14, 15. Con la implementación de estas historias de usuario se gestionará todo lo referente al control de acceso en la aplicación y la seguridad de la misma. Estas historias de usuario corresponden a los procesos de gestionar las cuentas dentro de la aplicación para los diferentes usuarios así como la autenticación de dichos usuarios. Con la culminación de la iteración ya se tendrá la aplicación lista para, en conjunto con las funcionalidades obtenidas de las demás iteraciones, 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 18. Plan de duración de las iteraciones

Iteración	Historias de Usuario	Duración total iteraciones
Iteración 1	Gestionar deporte	6 semanas
	Gestionar delegación	
	Conformar edición de Juegos Deportivos	
	Gestionar equipo	
	Gestionar modalidad	
	Gestionar división	
Iteración 2	Gestionar competencia	6 semanas
	Gestionar grupo	
	Gestionar enfrentamiento	
	Gestionar evento	
	Gestionar resultados	
Iteración 3	Confeccionar tabla de posición	2 semanas
	Gestionar reclamación	
Iteración 4	Gestionar usuario	2 semanas
	Autenticar	

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 19. Módulos e HU abarcadas

Módulos	Historias de Usuario
Deportes	Gestionar deporte
	Gestionar modalidad

	Gestionar división
Delegaciones	Gestionar delegación
	Gestionar equipo
Ediciones	Conformar edición de Juegos Deportivos
Competencias	Gestionar competencia
	Gestionar grupo
Programa Competencia	Gestionar enfrentamiento
	Gestionar evento
Resultados	Gestionar resultados
Emulación	Confeccionar tabla de posición
	Gestionar reclamación
Usuario	Gestionar usuario
	Autenticar

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 20. Plan de duración de entregas

Módulos	Final iter 1 4ta sem. febrero	Final iter 2 4ta sem. marzo	Final iter 3 4ta sem. abril	Final iter 4 2da sem. mayo
Deportes	0.1	1.0	Finalizado	Finalizado
Delegaciones	0.1	1.0	Finalizado	Finalizado
Ediciones	0.1	1.0	Finalizado	Finalizado
Competencias		0.1	0.2	1.0
Programa Competencia		0.1	0.2	1.0
Resultado		0.1	0.2	1.0
Emulación				1.0
Usuario				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.

Capítulo 3 Implementación y Prueba

3.1. Introducción

Es un secreto a voces el decir que el momento cumbre del desarrollo de un proyecto lo constituye 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 inherentes 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 21. Tarjeta CRC clase Competencia

Competencia	
Funcionalidades	Colaboraciones
Crear una competencia	Grupo
Eliminar una competencia	Evento
Devolver el valor de un atributo	Deporte
Cambiar el valor de un atributo	Recuperación
Devolver el deporte al que pertenece la competencia	
Cambiar el deporte al que pertenece la competencia	
Devolver los evento que pertenecen a la competencia	
Adicionar eventos a la competencia	
Devolver los grupos que pertenecen a la competencia	
Adicionar grupos a la competencia	
Devolver los equipos disponibles para una competencia	
Devolver los eventos que pertenecen a la competencia	

Tabla 22. Tarjeta CRC clase Deporte

Deporte	
Funcionalidades	Colaboraciones
Crear un deporte	Competencia Equipo Modalidad
Eliminar un deporte	
Devolver el valor de un atributo	
Cambiar el valor de un atributo	
Devolver las competencias asociadas al deporte	
Devolver los equipos asociados al deporte	
Devolver las modalidades asociadas al deporte	

Tabla 23. Tarjeta CRC clase Enfrentamiento

Enfrentamiento	
Funcionalidades	Colaboraciones
Crear un enfrentamiento	Evento Equipo
Eliminar un enfrentamiento	
Devolver el valor de un atributo	
Cambiar el valor de un atributo	
Introducir resultado del enfrentamiento	

Tabla 24. Tarjeta CRC clase Equipo

Equipo	
Funcionalidades	Colaboraciones
Crear un equipo	Deporte Grupo JuegoFacultad
Eliminar un equipo	
Devolver el valor de un atributo	
Cambiar el valor de un atributo	
Devolver el deporte al que pertenece el equipo	
Devolver el grupo al que pertenece el equipo	
Cambiar el grupo al que pertenece el equipo	
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 25. Tarjeta CRC clase Evento

Evento	
Funcionalidades	Colaboraciones
Crear un evento	Competencia
Eliminar un evento	EventoModalidad
Devolver el valor de un atributo	Enfrentamiento
Cambiar el valor de un atributo	
Devolver la competencia a la que pertenece el evento	
Devolver el enfrentamiento asociado a este evento	
Devolver el evento por modalidad asociado a este evento	
Devolver los atletas que han sido ganadores en un evento	
Verificar los enfrentamientos creados	

Tabla 26. Tarjeta CRC clase EventoModalidad

EventoModalidad	
Funcionalidades	Colaboraciones
Crear un evento por modalidad	Evento
Eliminar un evento por modalidad	Modalidad
Devolver el valor de un atributo	
Cambiar el valor de un atributo	
Devolver el evento por modalidad	
Introducir o modificar los resultados del evento por modalidad	

Tabla 27. Tarjeta CRC clase EventoDivision

EventoDivision	
Funcionalidades	Colaboraciones
Crear un evento por división	Evento
Eliminar un evento por división	Modalidad
Devolver el valor de un atributo	
Cambiar el valor de un atributo	
Devolver el evento por división	
Introducir o modificar los resultados del evento por división	

Tabla 28. Tarjeta CRC clase Facultad

Facultad	
Funcionalidades	Colaboraciones
Crear una facultad	JuegoFacultad
Eliminar una facultad	
Devolver el valor de un atributo	
Cambiar el valor de un atributo	
Devolver los juegos donde estuvo presente la facultad	

Tabla 29. Tarjeta CRC clase Grupo

Grupo	
Funcionalidades	Colaboraciones
Crear un evento	Equipo
Eliminar un evento	Competencia
Devolver el valor de un atributo	
Cambiar el valor de un atributo	
Devolver la competencia a la que pertenece este grupo	
Devolver los equipos que pertenecen a este grupo	
Adicionar equipos al grupo	
Eliminar equipos del grupo	
Devolver los equipos que no estén en grupo	

Verificar que un grupo exista para una competencia determinada	
--	--

Tabla 30. Tarjeta CRC clase Juego

Juego	
Funcionalidades	Colaboraciones
Crear una edición de juego	JuegoFacultad
Devolver el valor de un atributo	Competencia
Cambiar el valor de un atributo	Reclamacion
Devolver las competencias asociadas a esta edición de juego	
Adicionar una competencia a la presente edición de juego	
Devolver las facultades presentes en esta edición ordenadas por su puntuación.	
Adicionar una facultad a la presente edición de juego	
Devolver la reclamaciones asociadas a la edición de juego	
Adicionar reclamaciones a la presente edición de juego	
Devolver las facultades presentes en esta edición ordenadas por sus medallas	

Tabla 31. Tarjeta CRC clase JuegoFacultad

JuegoFacultad	
Funcionalidades	Colaboraciones
Crear representación de una facultad en una edición de juego	Juego Facultad
Eliminar representación de una facultad en una edición de juego	Equipo
Devolver el valor de un atributo	
Cambiar el valor de un atributo	
Devolver los equipos de una facultad asociados a una	

edición de juego	
Adicionar equipo a una facultad en una edición de juego	
Devolver las medallas obtenidas por una facultad en una edición de juego	
Adicionar o eliminar medallas a una facultad en una edición de juego	
Conocer la posición final de una delegación en una edición de juego	
Devolver la puntuación obtenida por una facultad en una edición de juego	
Adicionar o eliminar puntos a una facultad en una edición de juego	

Tabla 32. Tarjeta CRC clase Modalidad

Modalidad	
Funcionalidades	Colaboraciones
Crear una modalidad	Deporte EventoModalidad
Eliminar una modalidad	
Devolver el valor de un atributo	
Cambiar el valor de un atributo	
Devolver el deporte de la modalidad	
Devolver los eventos por modalidad a los que esta asociado esta modalidad	

Tabla 33. Tarjeta CRC clase Reclamación

Reclamación	
Funcionalidades	Colaboraciones
Crear una reclamación	Juego
Eliminar una reclamación	
Devolver el valor de un atributo	
Cambiar el valor de un atributo	

Devolver la edición de juego a la que esta asociada la reclamación	
--	--

Tabla 34. Tarjeta CRC clase División

División	
Funcionalidades	Colaboraciones
Crear una división	Deporte EventoDivision
Eliminar una división	
Devolver el valor de un atributo	
Cambiar el valor de un atributo	
Devolver el deporte de la división	
Devolver los eventos por división a los que esta asociado esta división	

Tabla 35. Tarjeta CRC clase Usuario

Usuario	
Funcionalidades	Colaboraciones
Crear un usuario	
Eliminar un usuario	
Devolver el valor de un atributo	
Cambiar el valor de un atributo	
Autenticar un usuario	

Tabla 36. Tarjeta CRC clase Atleta

Atleta	
Funcionalidades	Colaboraciones
Crear un atleta	AtletaEvento
Eliminar un atleta	
Devolver el valor de un atributo	
Cambiar el valor de un atributo	

Tabla 37. Tarjeta CRC clase AtletaEvento

AtletaEvento	
Funcionalidades	Colaboraciones
Asignar atleta ganador a un evento determinado	Atleta
Eliminar atleta ganador de un evento determinado	Evento
Devolver el valor de un atributo	
Cambiar el valor de un atributo	

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 Propel⁷ 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 38. Transformación ORM

Relacional	Orientado a objetos
Tabla	Clase
Fila	Objeto
Campos	Atributos

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.

⁷ <http://propel.phpdb.org/>

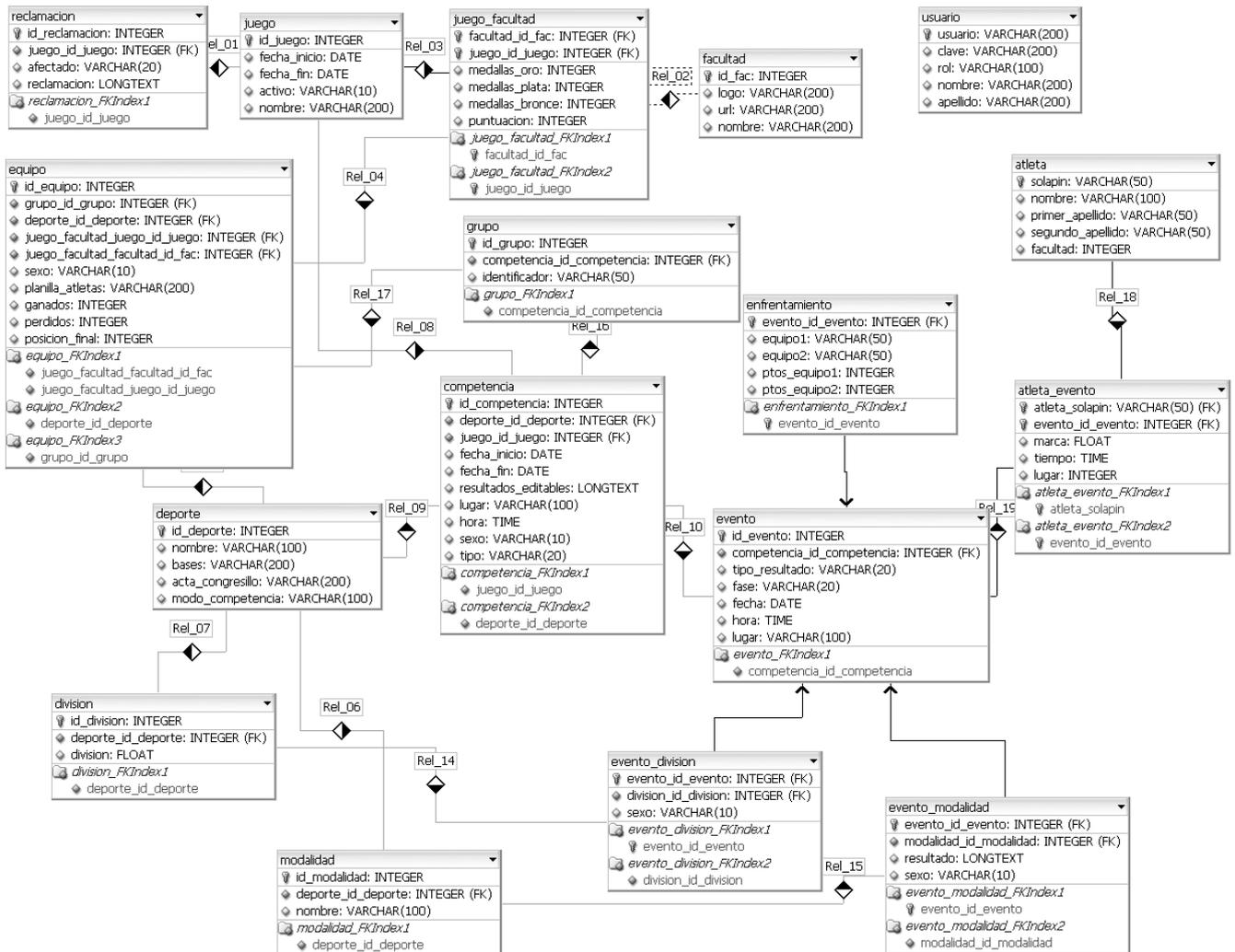


Figura 5. Modelo de datos

3.3. Implementación

Para llevar a cabo la correcta implementación de los módulos definidos en la propuesta de solución, con las consiguientes 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 39. Módulos abordados en la primera iteración

Módulo	Historia de usuario	Tiempo de Implementación (semana)	
		Estimado	Real
Deporte	Gestionar deporte	1	0.50
	Gestionar modalidad	1	0.25
	Gestionar división	1	0.25
Delegación	Gestionar delegación	1	0.50
	Gestionar equipo	1	0.50
Edición	Conformar Edición JD	1	0.25

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

Módulo Deporte

Tabla 40. Tarea 1 módulo Deporte

Tarea	
Numero de tarea:1	Numero HU:1
Nombre de la tarea: Crear el módulo Deporte	
Tipo de tarea: Configuración	Punto de estimación:0.50
Fecha de inicio: 3 de febrero del 2009	Fecha de fin: 3 de febrero del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se crea el módulo Deporte, atendiendo a la tecnología usada (framework symfony).	

Tabla 41. Tarea 2 módulo Deporte

Tarea	
Numero de tarea:2	Numero HU:1
Nombre de la tarea: Definir e implementar las acciones dentro del módulo para gestionar deportes.	

Tipo de tarea: Desarrollo	Punto de estimación: 0.50
Fecha de inicio: 3 de febrero del 2009	Fecha de fin: 3 de febrero del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se definen e implementan las acciones que se realizarán dentro del módulo Deporte. Aquí se implementarán las acciones para crear, modificar y eliminar un deporte.	

Tabla 42. Tarea 3 módulo Deporte

Tarea	
Numero de tarea: 3	Numero HU: 1
Nombre de la tarea: Definir e implementar las acciones dentro del módulo para gestionar modalidades y divisiones.	
Tipo de tarea: Desarrollo	Punto de estimación: 0.50
Fecha de inicio: 3 de febrero del 2009	Fecha de fin: 3 de febrero del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se definen las acciones que se realizarán dentro del módulo Deporte relacionadas con las modalidades y divisiones pertenecientes a un deporte. Aquí se definen e implementan las acciones para crear, modificar y eliminar una modalidad o división a un deporte dado.	

Tabla 43. Tarea 4 módulo Deporte

Tarea	
Numero de tarea: 4	Numero HU: 1
Nombre de la tarea: Diseñar las plantillas para cada acción del módulo deporte	
Tipo de tarea: Desarrollo	Punto de estimación: 0.50
Fecha de inicio: 4 de febrero del 2009	Fecha de fin: 4 de febrero del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se diseña las plantillas para cada una de las acciones del módulo Deporte.	

Módulo Delegación

Tabla 44. Tarea 1 módulo Delegación

Tarea	
Numero de tarea: 1	Numero HU: 2,4
Nombre de la tarea: Crear el módulo Delegación	

Tipo de tarea: Configuración	Punto de estimación: 0.50
Fecha de inicio: 9 de febrero del 2009	Fecha de fin: 9 de febrero del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se crea el módulo Delegación, atendiendo a la tecnología usada (framework symfony).	

Tabla 45. Tarea 2 módulo Delegación

Tarea	
Numero de tarea: 2	Numero HU: 2
Nombre de la tarea: Definir e implementar las acciones dentro del módulo para las delegaciones	
Tipo de tarea: Desarrollo	Punto de estimación: 0.50
Fecha de inicio: 9 de febrero del 2009	Fecha de fin: 9 de febrero del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se definen e implementan las acciones dentro del módulo para las delegaciones. Aquí se van a implementar las acciones para crear, modificar y eliminar delegaciones.	

Tabla 46. Tarea 3 módulo Delegación

Tarea	
Numero de tarea: 2	Numero HU: 4
Nombre de la tarea: Definir e implementar las acciones dentro del módulo Delegación para los equipos.	
Tipo de tarea: Desarrollo	Punto de estimación: 0.60
Fecha de inicio: 23 de febrero del 2009	Fecha de fin: 23 de febrero del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se definen e implementan las acciones dentro del módulo Delegación para crear, modificar y eliminar un equipo de una delegación.	

Tabla 47. Tarea 3 módulo Delegación

Tarea	
Numero de tarea: 3	Numero HU: 2,4
Nombre de la tarea: Diseñar las plantillas para cada acción dentro del módulo	
Tipo de tarea: Desarrollo	Punto de estimación: 0.50

Fecha de inicio: 10 de febrero del 2009	Fecha de fin: 10 de febrero del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se diseña las plantillas para cada una de las acciones del módulo Delegación. Este diseño se va a realizar con las diferentes etiquetas definidas por framework para ello.	

Módulo Edición

Tabla 48. Tarea 1 módulo Edición

Tarea	
Numero de tarea: 1	Numero HU: 3
Nombre de la tarea: Crear el módulo Edición	
Tipo de tarea: Configuración	Punto de estimación: 0.25
Fecha de inicio: 16 de febrero del 2009	Fecha de fin: 16 de febrero del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se crea el módulo Edición, atendiendo a la tecnología usada (framework symfony).	

Tabla 49. Tarea 2 módulo Edición

Tarea	
Numero de tarea: 2	Numero HU: 3
Nombre de la tarea: Definir e implementar las acciones dentro del módulo Edición.	
Tipo de tarea: Desarrollo	Punto de estimación: 0.25
Fecha de inicio: 16 de febrero del 2009	Fecha de fin: 16 de febrero del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se definen e implementan las acciones del módulo Edición para crear y modificar una edición de los juegos deportivos.	

Tabla 50. Tarea 1 módulo Edición

Tarea	
Numero de tarea: 3	Numero HU: 3
Nombre de la tarea: Diseñar las plantillas para cada acción del módulo Edición.	
Tipo de tarea: Desarrollo	Punto de estimación: 0.25
Fecha de inicio: 17 de febrero del 2009	Fecha de fin: 18 de febrero del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	

Descripción: Se diseñan las plantillas para cada una de las acciones del módulo Edición. Aquí se va a definir como se van a mostrar las plantillas para las acciones crear, modificar una edición de los juegos deportivos.

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 51. Módulos abordados en la segunda iteración

Módulo	Historia de usuario	Tiempo de Implementación (semana)	
		Estimado	Real
Competencia	Gestionar competencia	1	0.70
	Gestionar grupo	1	0.75
Programa	Gestionar enfrentamiento	1	0.50
Competencia	Gestionar evento	1	0.50
Resultados	Gestionar resultados	2	2

Módulo Competencia

Tabla 52. Tarea 1 módulo Competencia

Tarea	
Numero de tarea:1	Numero HU:5,6
Nombre de la tarea: Crear el módulo Competencia	
Tipo de tarea: Configuración	Punto de estimación: 0.70
Fecha de inicio: 2 de marzo del 2009	Fecha de fin: 2 de marzo del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se crea el módulo competencia, según la tecnología usada.	

Tabla 53. Tarea 2 módulo Competencia

Tarea	
Numero de tarea:2	Numero HU:5
Nombre de la tarea: Definir e implementar las acciones dentro del módulo para las competencias	

Tipo de tarea: Desarrollo	Punto de estimación: 0.70
Fecha de inicio: 2 de marzo del 2009	Fecha de fin: 2 de marzo del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se definen e implementan las acciones dentro del módulo para crear, modificar y eliminar una competencia dentro del sistema.	

Tabla 54. Tarea 3 módulo Competencia

Tarea	
Numero de tarea: 3	Numero HU: 6
Nombre de la tarea: Definir e implementar las acciones dentro del módulo para los grupos.	
Tipo de tarea: Desarrollo	Punto de estimación: 0.70
Fecha de inicio: 3 de marzo del 2009	Fecha de fin: 3 de febrero del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se define e implementan dentro del módulo Competencia las acciones relacionadas con los grupos que pertenecen a una competencia. Aquí se van a programar una acción para crear, modificar y eliminar un grupo.	

Tabla 55. Tarea 4 módulo Competencia

Tarea	
Numero de tarea: 4	Numero HU: 5,6
Nombre de la tarea: Diseñar las plantillas para cada acción del módulo competencia.	
Tipo de tarea: Desarrollo	Punto de estimación: 0.70
Fecha de inicio: 4 de marzo del 2009	Fecha de fin: 4 de marzo del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se diseña las plantillas para cada una de las acciones del módulo Competencia. Se van a definir como se van a mostrar las plantillas para las acciones.	

Módulo Programa de Competencia

Tabla 56. Tarea 1 módulo Programa Competencia

Tarea	
Numero de tarea: 1	Numero HU: 7,8

Nombre de la tarea: Crear el módulo Programa de Competencia	
Tipo de tarea: Configuración	Punto de estimación: 0.75
Fecha de inicio: 9 de marzo del 2009	Fecha de fin: 9 de marzo del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se crea el módulo Programa Competencia, atendiendo a la tecnología usada (framework symfony).	

Tabla 57. Tarea 2 módulo Programa Competencia

Tarea	
Numero de tarea: 2	Numero HU: 7
Nombre de la tarea: Definir e implementan las acciones dentro del módulo Programa Competencia para los enfrentamientos.	
Tipo de tarea: Desarrollo	Punto de estimación: 0.75
Fecha de inicio: 9 de marzo del 2009	Fecha de fin: 9 de marzo del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se definen e implementan las acciones dentro del módulo para crear, modificar y eliminar un enfrentamiento entre dos equipos.	

Tabla 58. Tarea 3 módulo Programa Competencia

Tarea	
Numero de tarea: 3	Numero HU: 8
Nombre de la tarea: Definir e implementan las acciones dentro del módulo Programa de Competencia para los eventos.	
Tipo de tarea: Desarrollo	Punto de estimación: 0.75
Fecha de inicio: 9 de marzo del 2009	Fecha de fin: 9 de marzo del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se definen e implementan dentro del módulo las acciones relacionadas con los eventos que pertenecen a una competencia. Aquí se van a definir una acción para crear, modificar y eliminar un evento.	

Tabla 59. Tarea 4 módulo Programa Competencia

Tarea	
Numero de tarea: 4	Numero HU: 7,8
Nombre de la tarea: Diseñar las plantillas para cada acción del módulo Programa Competencia.	
Tipo de tarea: Desarrollo	Punto de estimación: 0.70
Fecha de inicio: 10 de marzo del 2009	Fecha de fin: 11 de marzo del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se diseña las plantillas para cada una de las acciones del módulo Programa Competencia.	

Módulo Resultado

Tabla 60. Tarea 1 módulo Resultado

Tarea	
Numero de tarea: 1	Numero HU: 9
Nombre de la tarea: Crear el módulo Resultado	
Tipo de tarea: Configuración	Punto de estimación: 2
Fecha de inicio: 16 de marzo del 2009	Fecha de fin: 16 de marzo del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se crea el módulo Resultado, atendiendo a la tecnología usada (framework symfony).	

Tabla 61. Tarea 2 módulo Resultado

Tarea	
Numero de tarea: 2	Numero HU: 9
Nombre de la tarea: Definir e implementar las acciones dentro del módulo Resultado	
Tipo de tarea: Desarrollo	Punto de estimación: 2
Fecha de inicio: 16 de marzo del 2009	Fecha de fin: 17 de marzo del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se definen las acciones del módulo Resultado. Aquí se van a definir una acción para crear y modificar un resultado tanto de una competencia en general como de un enfrentamiento o un evento.	

Tabla 62. Tarea 3 módulo Resultado

Tarea	
Numero de tarea: 3	Numero HU: 9
Nombre de la tarea: Diseñar las plantillas para cada acción del módulo Resultado	
Tipo de tarea: Desarrollo	Punto de estimación: 2
Fecha de inicio: 18 de marzo del 2009	Fecha de fin: 19 de marzo del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se diseña las plantillas para cada una de las acciones del módulo Resultado. Se van a definir como se van a mostrar las plantillas para las diferentes acciones que se van a realizar en dicho módulo.	

3.3.3. Iteración 3

Como parte de la presente iteración se desarrollaron las historias de usuario asociadas a los procesos de la emulación dentro de los juegos deportivos.

Tabla 63. Módulos abordados en la tercera iteración

Módulo	Historia de usuario	Tiempo de Implementación (semana)	
		Estimado	Real
Emulación	Confeccionar tabla posiciones	1	0.40
	Gestionar reclamación	1	0.40

Módulo Emulación

Tabla 64. Tarea 1 módulo Emulación

Tarea	
Numero de tarea: 1	Numero HU: 10, 11
Nombre de la tarea: Crear el módulo Emulación	
Tipo de tarea: Configuración	Punto de estimación: 0.40
Fecha de inicio: 31 de marzo del 2009	Fecha de fin: 31 de marzo del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se crea el módulo Emulación, atendiendo a la tecnología usada (framework symfony).	

Tabla 65. Tarea 2 módulo Emulación

Tarea	
Numero de tarea: 2	Numero HU: 10
Nombre de la tarea: Definir e implementar las acciones dentro del módulo Emulación para la confección de la tabla de posiciones.	
Tipo de tarea: Desarrollo	Punto de estimación: 0.40
Fecha de inicio: 31 de marzo del 2009	Fecha de fin: 31 de marzo del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se definen las acciones dentro del módulo Emulación para añadir y modificar medallas o puntos.	

Tabla 66. Tarea 3 módulo Emulación

Tarea	
Numero de tarea: 3	Numero HU: 11
Nombre de la tarea: Definir e implementar las acciones dentro del módulo Emulación para realizar reclamaciones.	
Tipo de tarea: Desarrollo	Punto de estimación: 0.40
Fecha de inicio: 31 de marzo del 2009	Fecha de fin: 31 de marzo del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se definen las acciones dentro del módulo Emulación para realizar reclamaciones. Aquí se van a definir las acciones crear, modificar o eliminar una reclamación a la presente edición de juegos deportivos.	

Tabla 67. Tarea 4 módulo Emulación

Tarea	
Numero de tarea: 4	Numero HU: 10, 11
Nombre de la tarea: Diseñar las plantillas para cada acción del módulo Emulación	
Tipo de tarea: Desarrollo	Punto de estimación: 0.40
Fecha de inicio: 1 de abril del 2009	Fecha de fin: 1 de abril del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se van a definir como se van a mostrar las plantillas para las diferentes acciones que se van a realizar dentro del módulo Emulación.	

3.3.4. Iteración 4

Las historias de usuario implementadas en esta iteración responden a los requerimientos de seguridad presentes en el sistema.

Tabla 68. Módulos abordados en la cuarta iteración

Módulo	Historia de usuario	Tiempo de Implementación (semana)	
		Estimado	Real
Usuario	Gestionar usuario	1	0.20
	Autenticar	1	0.15

Módulo Usuario

Tabla 69. Tarea 1 módulo Usuario

Tarea	
Numero de tarea: 1	Numero HU: 14
Nombre de la tarea: Crear el módulo Usuario	
Tipo de tarea: Configuración	Punto de estimación: 0.20
Fecha de inicio: 7 de abril del 2009	Fecha de fin: 7 de abril del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se crea el módulo atendiendo a la tecnología usada (framework symfony).	

Tabla 70. Tarea 2 módulo Usuario

Tarea	
Numero de tarea: 2	Numero HU: 14
Nombre de la tarea: Definir e implementar las acciones dentro del módulo Usuario	
Tipo de tarea: Desarrollo	Punto de estimación: 0.20
Fecha de inicio: 7 de abril del 2009	Fecha de fin: 7 de abril del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se definen e implementan las acciones dentro del módulo Usuario para crear, modificar o eliminar un usuario dentro del sistema. Así como para autenticar dicho usuario.	

Tabla 71. Tarea 1 módulo Usuario

Tarea	
Numero de tarea: 3	Numero HU: 14
Nombre de la tarea: Diseñar las plantillas para cada acción del módulo Usuario	
Tipo de tarea: Desarrollo	Punto de estimación: 0.20
Fecha de inicio: 8 de abril del 2009	Fecha de fin: 8 de abril del 2009
Programador responsable: Michel Legra – Elizabet Rodríguez	
Descripción: Se diseña las plantillas para cada una de las acciones del módulo. Aquí se van a definir como se van a mostrar las plantillas para las diferentes acciones que se van a realizar en él.	

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.

Para el desarrollo de las pruebas del presente trabajo se realizó una combinación de pruebas manuales con pruebas automatizadas toda vez que se consideró que ambos tipos se complementan de manera eficiente logrando así una mayor calidad de la aplicación.

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 72. Prueba 1 del módulo Deporte

Caso de prueba de aceptación	
Código: HU1_P1	Historia de Usuario: 1
Nombre: Adicionar un deporte al sistema.	
Descripción: Prueba para la funcionalidad de adicionar un deporte al sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos.	
Entradas / Pasos de ejecución: Para la incorporación de un deporte al sistema se deben introducir los datos necesarios para conformar el mismo.	
Resultados esperados: El deporte es incorporado al sistema sin que se generen errores.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 73. Prueba 2 del módulo Deporte

Caso de prueba de aceptación	
Código: HU1_P2	Historia de Usuario: 1
Nombre: Modificar un deporte del sistema.	
Descripción: Prueba para la funcionalidad de modificar un deporte que se encuentre inscrito en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema.	

Se usará un usuario con datos válidos. El deporte que se quiere modificar debe de estar inscrito en el sistema.
Entradas / Pasos de ejecución: Para modificar un deporte del sistema se debe seleccionar uno de los deportes previamente inscritos en él, después se deben mostrar los datos referentes a dicho deporte posibilitando que e puedan modificarse los que se estimen convenientes.
Resultados esperados: Se realizan las modificaciones al deporte seleccionado sin que se generen errores.
Evaluación de la prueba: Prueba satisfactoria

Tabla 74. Prueba 3 del módulo Deporte

Caso de prueba de aceptación	
Código: HU1_P3	Historia de Usuario: 1
Nombre: Eliminar un deporte del sistema.	
Descripción: Prueba para la funcionalidad de eliminar un deporte que se encuentre inscrito en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. El deporte que se seleccione debe estar inscrito en el sistema.	
Entradas / Pasos de ejecución: Para eliminar un deporte del sistema se debe seleccionar uno de los deportes previamente inscritos en él, luego se procederá a eliminar dicho deporte.	
Resultados esperados: Se elimina el deporte seleccionado sin que se generen errores.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 75. Prueba 1 del módulo Delegación

Caso de prueba de aceptación	
Código: HU2_P1	Historia de Usuario: 2
Nombre: Adicionar una delegación al sistema.	
Descripción: Prueba para la funcionalidad de adicionar una delegación al sistema.	

Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos.
Entradas / Pasos de ejecución: Para la incorporación de una delegación al sistema se deben introducir los datos necesarios para conformar la misma.
Resultados esperados: La delegación es incorporada al sistema sin que se generen errores.
Evaluación de la prueba: Prueba satisfactoria

Tabla 76. Prueba 2 del módulo Delegación

Caso de prueba de aceptación	
Código: HU2_P2	Historia de Usuario: 2
Nombre: Modificar una delegación del sistema.	
Descripción: Prueba para la funcionalidad de modificar una delegación que se encuentre inscrita en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. La delegación que se quiere modificar debe de estar inscrita en el sistema.	
Entradas / Pasos de ejecución: Para modificar una delegación del sistema se debe seleccionar una de las delegaciones previamente inscritas en él, después se debe mostrar los datos referentes a dicha delegación posibilitando que puedan modificarse los que se estimen convenientes.	
Resultados esperados: Se realizan las modificaciones a la delegación seleccionada sin que se generen errores.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 77. Prueba 3 del módulo Delegación

Caso de prueba de aceptación	
Código: HU2_P3	Historia de Usuario: 2

Nombre: Eliminar una delegación del sistema.
Descripción: Prueba para la funcionalidad de eliminar una delegación que se encuentre inscrita en el sistema.
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. La delegación que se quiere eliminar debe de estar inscrita en el sistema.
Entradas / Pasos de ejecución: Para eliminar una delegación del sistema se debe seleccionar una de las delegaciones previamente inscritas en él, luego se procederá a eliminar dicha delegación.
Resultados esperados: Se elimina la delegación seleccionada sin que se generen errores.
Evaluación de la prueba: Prueba satisfactoria

Tabla 78. Prueba 4 del módulo Delegación

Caso de prueba de aceptación	
Código: HU4_P4	Historia de Usuario: 4
Nombre: Adicionar un equipo al sistema.	
Descripción: Prueba para la funcionalidad de adicionar equipo de un deporte a una delegación presente en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. Debe estar inscrito el deporte del cual se quiere crear el equipo. Debe estar inscrita la delegación de la cual se quiere crear el equipo.	
Entradas / Pasos de ejecución: Para la incorporación de un equipo al sistema se debe introducir los datos necesarios para conformar el mismo.	
Resultados esperados: El equipo es incorporado al sistema sin que se generen errores.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 79. Prueba 5 del módulo Delegación

Caso de prueba de aceptación	
Código: HU4_P5	Historia de Usuario: 4
Nombre: Modificar un equipo del sistema.	
Descripción: Prueba para la funcionalidad de modificar equipo que se encuentre inscrito en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. El equipo que se quiere modificar, debe estar inscrito en el sistema.	
Entradas / Pasos de ejecución: Para modificar un equipo dentro del sistema se debe seleccionar un equipo de un deporte determinado, después se deben mostrar los datos referentes a dicho equipo para que se puedan modificar los que se estimen convenientes.	
Resultados esperados: Se modifican los datos relacionados con el equipo escogido.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 80. Prueba 6 del módulo Delegación

Caso de prueba de aceptación	
Código: HU4_P6	Historia de Usuario: 4
Nombre: Eliminar un equipo del sistema.	
Descripción: Prueba para la funcionalidad de eliminar un equipo que se encuentre inscrito en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. El equipo que se quiere eliminar debe estar inscrito en el sistema.	
Entradas / Pasos de ejecución: Para eliminar un equipo del sistema se debe seleccionar uno de los equipos previamente inscritos en él, luego se procederá a eliminar dicho equipo.	
Resultados esperados:	

Se elimina el equipo seleccionado.
Evaluación de la prueba: Prueba satisfactoria

Tabla 81. Prueba 7 del módulo Delegación

Caso de prueba de aceptación	
Código: HU4_P7	Historia de Usuario: 4
Nombre: Adicionar equipo sin que se encuentren un deporte inscritos.	
Descripción: Prueba para la funcionalidad de adicionar un equipo al sistema sin que se inserte con anterioridad algún deporte.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos.	
Entradas / Pasos de ejecución: Si el usuario quiere adicionar un equipo al sistema y no se encuentran adicionados previamente los deportes al sistema.	
Resultados esperados: Se muestra un mensaje de error para indicar que se debe inscribir algún deporte.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 82. Prueba 8 del módulo Delegación

Caso de prueba de aceptación	
Código: HU4_P8	Historia de Usuario: 4
Nombre: Adicionar equipo que ya se encuentre inscrito.	
Descripción: Prueba para la funcionalidad de adicionar un equipo que ya se encuentre inscrito dentro del sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos.	
Entradas / Pasos de ejecución: Si el usuario quiere adicionar un equipo al sistema y este ya se encuentra adicionado al mismo.	
Resultados esperados:	

Se muestra un mensaje de error para indicarle que ya se ha inscrito un equipo de esta delegación en este deporte.
Evaluación de la prueba: Prueba satisfactoria

Tabla 83. Prueba 1 del módulo Edición

Caso de prueba de aceptación	
Código: HU3_P1	Historia de Usuario: 3
Nombre: Crear una edición de juegos deportivos.	
Descripción: Prueba para la funcionalidad de crear una nueva edición de los juegos deportivos.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. Las delegaciones deben estar previamente inscritas en el sistema.	
Entradas / Pasos de ejecución: Para la incorporación de una nueva edición de juegos deportivos al sistema se deben introducir los datos necesarios para conformar los mismos.	
Resultados esperados: La nueva edición de los juegos deportivos es incorporada al sistema sin que se generen errores.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 84. Prueba 2 del módulo Edición

Caso de prueba de aceptación	
Código: HU3_P2	Historia de Usuario: 3
Nombre: Modificar una edición de los juegos deportivos dentro del sistema.	
Descripción: Prueba para la funcionalidad de modificar la ultima edición de los juegos deportivos que se encuentre inscrita en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. El juegos deportivo que se quiere modificar, debe estar inscritos en el sistema y debe ser el ultimo (el que se este jugando).	

Entradas / Pasos de ejecución:
Para modificar unos juegos deportivos el sistema debe mostrar la última edición de juegos deportivos que existan por lo tanto solo se dará la opción de modificar la última edición creada. Se debe mostrar los datos referentes a dicha edición de juegos deportivos para que se puedan modificar los datos que se estimen convenientes.
Resultados esperados:
Se modifican los datos relacionados con la última edición de juegos deportivos.
Evaluación de la prueba: Prueba satisfactoria

Tabla 85. Prueba 1 del módulo Competencia

Caso de prueba de aceptación	
Código: HU5_P1	Historia de Usuario: 5
Nombre: Adicionar una competencia al sistema.	
Descripción: Prueba para la funcionalidad de adicionar competencia al sistema.	
Condiciones de Ejecución:	
El usuario debe estar autenticado en el sistema.	
Se usará un usuario con datos válidos.	
Debe estar inscrito el deporte del cual se quiere crear la competencia.	
Deben estar inscritas las delegaciones.	
Entradas / Pasos de ejecución:	
Para la incorporación de una nueva competencia al sistema se debe introducir los datos necesarios para conformar la mismo especificando el tipo de competencia que se quiera realizar.	
Resultados esperados:	
La competencia es incorporada al sistema sin que se generen errores.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 86. Prueba 2 del módulo Competencia

Caso de prueba de aceptación	
Código: HU5_P2	Historia de Usuario: 5
Nombre: Modificar una competencia.	
Descripción: Prueba para la funcionalidad de modificar una competencia que se encuentre inscrita en el sistema.	

<p>Condiciones de Ejecución:</p> <p>El usuario debe estar autenticado en el sistema.</p> <p>Se usará un usuario con datos válidos.</p> <p>La competencia debe estar inscrita en el sistema.</p>
<p>Entradas / Pasos de ejecución:</p> <p>Para modificar una competencia dentro del sistema se deben mostrar las competencias que han sido creadas para que el usuario seleccione la que desea modificar, después se deben mostrar los datos referentes a dicha competencia para que puedan modificarse los datos deseados.</p>
<p>Resultados esperados:</p> <p>Se modifican los datos relacionados con la competencia seleccionada.</p>
<p>Evaluación de la prueba: Prueba satisfactoria</p>

Tabla 87. Prueba 3 del módulo Competencia

Caso de prueba de aceptación	
Código: HU5_P3	Historia de Usuario: 5
Nombre: Eliminar una competencia del sistema.	
Descripción: Prueba para la funcionalidad de eliminar una competencia que se encuentre inscrita en el sistema.	
<p>Condiciones de Ejecución:</p> <p>El usuario debe estar autenticado en el sistema.</p> <p>Se usará un usuario con datos válidos.</p> <p>La competencia que se quiera eliminar debe estar inscrita en el sistema.</p>	
<p>Entradas / Pasos de ejecución:</p> <p>Para eliminar una competencia del sistema se le debe mostrar todas las competencias previamente inscritas en el sistema, luego se procederá a eliminar la competencia escogida por el usuario.</p>	
<p>Resultados esperados:</p> <p>Se elimina la competencia seleccionada.</p>	
<p>Evaluación de la prueba: Prueba satisfactoria</p>	

Tabla 88. Prueba 4 del módulo Competencia

Caso de prueba de aceptación	
Código: HU5_P4	Historia de Usuario: 5
Nombre: Adicionar una competencia con fecha fuera de rango de los juegos deportivos.	
Descripción: Prueba para la funcionalidad de adicionar una competencia al sistema con fecha de duración fuera del rango de los juegos deportivos.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos.	
Entradas / Pasos de ejecución: Si el usuario quiere adicionar una competencia al sistema y la fecha de duración de la misma esta fuera del rango de los juegos deportivos en los se quiere inscribir la competencia.	
Resultados esperados: Se muestra un mensaje de error para indicar que la competencia debe estar en el rango de los juegos deportivos que se están efectuando.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 89. Prueba 5 del módulo Competencia

Caso de prueba de aceptación	
Código: HU5_P5	Historia de Usuario: 5
Nombre: Adicionar una competencia sin que se encuentren deportes inscritos.	
Descripción: Prueba para la funcionalidad de adicionar una competencia al sistema sin que se encuentre adicionado con anterioridad algún deporte.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos.	
Entradas / Pasos de ejecución: Si el usuario quiere adicionar una competencia al sistema y no se encuentran adicionados previamente deportes al sistema.	
Resultados esperados: Se muestra un mensaje de error para indicarle al administrador que debe inscribir algún deporte.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 90. Prueba 6 del módulo Competencia

Caso de prueba de aceptación	
Código: HU6_P6	Historia de Usuario: 6
Nombre: Adicionar un grupo al sistema.	
Descripción: Prueba para la funcionalidad de adicionar un grupo de una competencia al sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. Debe estar inscrita la competencia del cual se quiere crear el grupo y la misma debe ser de grupos. Deben estar inscritos como mínimo dos equipos del deporte al que pertenezca la competencia que se le quiere crear el grupo.	
Entradas / Pasos de ejecución: Para la incorporación de un nuevo grupo al sistema se debe introducir los datos necesarios para conformar le mismo, escogiendo con anterioridad la competencia de grupo de la cual se quiere crear el mismo.	
Resultados esperados: El grupo es incorporado al sistema sin que se generen errores.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 91. Prueba 7 del módulo Competencia

Caso de prueba de aceptación	
Código: HU6_P7	Historia de Usuario: 6
Nombre: Modificar un grupo.	
Descripción: Prueba para la funcionalidad de modificar un grupo de una competencia, que se encuentre inscrito en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. El grupo debe estar inscrito en el sistema.	
Entradas / Pasos de ejecución: Para modificar un grupo que este inscrito dentro del sistema se debe mostrar todos los grupos	

que han sido creados de la competencia escogida, para que el usuario pueda escoger el que desea modificar, después se deben mostrar los datos referentes a dicho grupo para que se modifiquen los deseados.
Resultados esperados: Se modifican los datos relacionados con el grupo escogido.
Evaluación de la prueba: Prueba satisfactoria

Tabla 92. Prueba 8 del módulo Competencia

Caso de prueba de aceptación	
Código: HU6_P8	Historia de Usuario: 6
Nombre: Eliminar un grupo del sistema.	
Descripción: Prueba para la funcionalidad de eliminar un grupo de una competencia que se encuentre inscrito en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. El grupo que se quiera eliminar debe estar inscrito en el sistema.	
Entradas / Pasos de ejecución: Para eliminar un grupo del sistema se deben mostrar todos los grupos previamente inscritos de una competencia en el sistema, luego se procederá a eliminar el grupo escogido por el usuario.	
Resultados esperados: Se elimina el grupo seleccionado.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 93. Prueba 9 del módulo Competencia

Caso de prueba de aceptación	
Código: HU6_P9	Historia de Usuario: 6
Nombre: Adicionar grupo sin que se encuentren equipos disponibles.	
Descripción: Prueba para la funcionalidad de adicionar un grupo al sistema sin que se encuentren disponible al menos dos equipos.	
Condiciones de Ejecución:	

El usuario debe de estar autenticado en el sistema. Se usará un usuario con datos válidos.
Entradas / Pasos de ejecución: Si el usuario quiere adicionar un grupo de una competencia de grupo determinada y no se encuentran al menos dos equipos disponibles para conformar dicho grupo.
Resultados esperados: Se muestra un mensaje de error para indicar que no existen equipos disponibles para conformar dicho grupo.
Evaluación de la prueba: Prueba satisfactoria

Tabla 94. Prueba 1 del módulo Programa Competencia

Caso de prueba de aceptación	
Código: HU7_P1	Historia de Usuario: 7
Nombre: Adicionar un enfrentamiento al sistema.	
Descripción: Prueba para la funcionalidad de adicionar un enfrentamiento entre dos equipos de una competencia determinada.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. Debe estar inscrita la competencia del cual se quiere crear el enfrentamiento y la misma debe ser de grupo. Deben estar inscritos los equipos del deporte al que pertenezca la competencia que se le quiere crear el enfrentamiento.	
Entradas / Pasos de ejecución: Para la incorporación de un nuevo enfrentamiento entre dos equipos al sistema se deben introducir los datos necesarios, escogiendo con anterioridad la competencia de grupo de la cual se quiere crear el mismo.	
Resultados esperados: El enfrentamiento es incorporado al sistema sin que se generen errores.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 95. Prueba 2 del módulo Programa Competencia

Caso de prueba de aceptación	
Código: HU7_P2	Historia de Usuario: 7
Nombre: Modificar un enfrentamiento.	
Descripción: Prueba para la funcionalidad de modificar un enfrentamiento entre dos equipos de una competencia, que se encuentre inscrito en el sistema.	
Condiciones de Ejecución: El usuario debe de estar autenticado en el sistema. Se usará un usuario con datos válidos. El enfrentamiento debe estar inscrito en el sistema.	
Entradas / Pasos de ejecución: Para modificar un enfrentamiento entre dos equipos se deben mostrar los enfrentamientos que han sido creados de la competencia escogida, para que el usuario pueda seleccionar el que desea modificar, después se deben mostrar los datos referentes a dicho enfrentamiento para que puedan modificarse los deseados.	
Resultados esperados: Se modifican los datos relacionados con el enfrentamiento escogido.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 96. Prueba 3 del módulo Programa Competencia

Caso de prueba de aceptación	
Código: HU7_P3	Historia de Usuario: 7
Nombre: Eliminar un enfrentamiento del sistema.	
Descripción: Prueba para la funcionalidad de eliminar un enfrentamiento entre dos equipos de una competencia que se encuentre inscrito en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. El enfrentamiento que se quiera eliminar debe estar inscrito en el sistema. El enfrentamiento no debe tener los resultados inscritos.	
Entradas / Pasos de ejecución: Para eliminar un enfrentamiento entre dos equipos del sistema se le debe mostrar todos los	

enfrentamientos previamente inscritos de la competencia seleccionada, con los diferentes datos que los identifiquen, luego se procederá a eliminar el enfrentamiento escogido por el usuario.
Resultados esperados: Se elimina el enfrentamiento seleccionado.
Evaluación de la prueba: Prueba satisfactoria

Tabla 97. Prueba 4 del módulo Programa Competencia

Caso de prueba de aceptación	
Código: HU6_P4	Historia de Usuario: 6
Nombre: Adicionar enfrentamiento sin que se encuentren inscritos equipos.	
Descripción: Prueba para la funcionalidad de adicionar un enfrentamiento al sistema sin que se encuentren inscritos dos equipo como mínimo dentro del sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. El enfrentamiento no debe tener los resultados inscritos.	
Entradas / Pasos de ejecución: Si el usuario quiere adicionar un enfrentamiento de una competencia determinada al sistema y no se encuentran inscritos dos o más equipos para conformarlo.	
Resultados esperados: Se muestra un mensaje de error para indicar que no se puede crear un enfrentamiento.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 98. Prueba 5 del módulo Programa Competencia

Caso de prueba de aceptación	
Código: HU6_P5	Historia de Usuario: 6
Nombre: Adicionar enfrentamiento existente en la misma fase.	
Descripción: Prueba para la funcionalidad de adicionar un enfrentamiento al sistema que ya se encuentre inscrito en la misma fase.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema.	

Se usará un usuario con datos válidos.
Entradas / Pasos de ejecución: Si el usuario quiere adicionar un enfrentamiento de una competencia determinada al sistema y este ya se encuentra inscrito en la misma fase.
Resultados esperados: Se muestra un mensaje de error para indicar que no se puede crear el enfrentamiento pues ya existe un enfrentamiento entre estos dos equipos en dicha fase.
Evaluación de la prueba: Prueba satisfactoria

Tabla 99. Prueba 6 del módulo Programa Competencia

Caso de prueba de aceptación	
Código: HU8_P6	Historia de Usuario: 8
Nombre: Adicionar un evento por modalidad al sistema.	
Descripción: Prueba para la funcionalidad de adicionar un evento por modalidad de una competencia de modalidad determinada.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. Debe de estar inscrita la competencia del cual se quiere crear el evento y la misma debe ser de modalidad. El deporte al que pertenezca la competencia a la cual si le quiera crear dicho evento debe tener modalidades inscritas.	
Entradas / Pasos de ejecución: Para la incorporación de un nuevo evento al sistema se debe introducir los datos necesarios, escogiendo con anterioridad la competencia de evento por modalidad de la cual se quiere crear el mismo.	
Resultados esperados: El evento por modalidad es incorporado al sistema sin que se generen errores.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 100. Prueba 7 del módulo Programa Competencia

Caso de prueba de aceptación	
Código: HU8_P7	Historia de Usuario: 8
Nombre: Modificar un evento por modalidad.	
Descripción: Prueba para la funcionalidad de modificar un evento por modalidad de una competencia que se encuentre inscrita en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. El evento por modalidad debe estar inscrito en el sistema. El evento por modalidad no debe tener inscritos sus resultados.	
Entradas / Pasos de ejecución: Para modificar un evento por modalidad que esté inscrito dentro del sistema se deben mostrar los eventos que han sido creados de la competencia escogida, para que el usuario pueda seleccionar el que desea modificar, después se deben mostrar los datos referentes a dicho evento para que se realicen los cambios deseados.	
Resultados esperados: Se modifican los datos relacionados con el evento por modalidad escogido.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 101. Prueba 8 del módulo Programa Competencia

Caso de prueba de aceptación	
Código: HU8_P8	Historia de Usuario: 8
Nombre: Eliminar un evento por modalidad del sistema.	
Descripción: Prueba para la funcionalidad de eliminar un evento por modalidad de una competencia que se encuentre inscrito en el sistema.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. El evento por modalidad que se quiera eliminar debe de estar inscrito en el sistema. El evento por modalidad no debe tener sus resultados inscritos.	
Entradas / Pasos de ejecución:	

Para eliminar un evento por modalidad del sistema se deben mostrar los eventos previamente inscritos de la competencia seleccionada, con los diferentes datos que los identifiquen, luego se procederá a eliminar el evento escogido por el usuario.
Resultados esperados: Se elimina el evento por modalidad seleccionado.
Evaluación de la prueba: Prueba satisfactoria

Tabla 102. Prueba 9 del módulo Programa Competencia

Caso de prueba de aceptación	
Código: HU8_P9	Historia de Usuario: 8
Nombre: Adicionar un evento por modalidad sin que se encuentren modalidades inscritas.	
Descripción: Prueba para la funcionalidad de adicionar un evento por modalidad sin que se encuentren inscritas previamente las modalidades del deporte de la competencia de dicho evento por modalidad.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. El evento por modalidad no debe tener sus resultados inscritos.	
Entradas / Pasos de ejecución: Si se desea adicionar un evento por modalidad y no se encuentran previamente inscritas las modalidades al deporte de la competencia de dicho evento.	
Resultados esperados: Se mostrará un mensaje de error indicando que deben de haber modalidades del deporte de esta competencia para poder adicionar un evento por modalidad a dicha competencia.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 103. Prueba 1 del módulo Resultado

Caso de prueba de aceptación	
Código: HU9_P1	Historia de Usuario: 9
Nombre: Introducir resultados al sistema.	
Descripción: Prueba para la funcionalidad de introducir resultados de una competencia	

determinada.
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. Debe estar inscrita la competencia del cual se quiere introducir el resultado.
Entradas / Pasos de ejecución: Para introducir los resultados de una competencia al sistema se debe rellenar los campos necesarios para la incorporación de los mismos, escogiendo con anterioridad la competencia a la cual se quiere introducir el resultado.
Resultados esperados: El resultado es incorporado al sistema sin que se generen errores.
Evaluación de la prueba: Prueba satisfactoria

Tabla 104. Prueba 2 del módulo Resultado

Caso de prueba de aceptación	
Código: HU8_P2	Historia de Usuario: 8
Nombre: Modificar resultados.	
Descripción: Prueba para la funcionalidad de modificar los resultado de una competencia.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. La competencia debe tener resultados previamente inscritos.	
Entradas / Pasos de ejecución: Para modificar el resultado de una competencia determinada dentro del sistema el usuario selecciona la misma, después se deben mostrar los resultados que se tenían almacenados de esta competencia para que se realicen los cambios deseados.	
Resultados esperados: Se modifican los resultados de la competencia escogida correctamente.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 105. Prueba 3 del módulo Resultado

Caso de prueba de aceptación	
Código: HU9_P3	Historia de Usuario:9
Nombre: Introducir resultados de un enfrentamiento.	
Descripción: Prueba para la funcionalidad de introducir resultados de un enfrentamiento de una competencia determinada.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos.	
Entradas / Pasos de ejecución: Para introducir los resultados de un enfrentamiento se selecciona el enfrentamiento en cuestión y se introduce la puntuación de ambos equipos.	
Resultados esperados: El resultado es incorporado al sistema sin que se generen errores.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 106. Prueba 3 del módulo Resultado

Caso de prueba de aceptación	
Código: HU9_P3	Historia de Usuario:9
Nombre: Introducir resultados de un evento por modalidad.	
Descripción: Prueba para la funcionalidad de introducir resultados de un evento por modalidad	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos.	
Entradas / Pasos de ejecución: Para introducir los resultados de un evento por modalidad se debe seleccionar el evento en cuestión y se introducen los datos necesarios para conformar dicho resultado.	
Resultados esperados: El resultado es incorporado al sistema sin que se generen errores.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 107. Prueba 1 del módulo Emulación

Caso de prueba de aceptación	
Código: HU10_P1	Historia de Usuario: 10
Nombre: Confeccionar medallero.	
Descripción: Prueba para la funcionalidad de confeccionar el medallero de la edición de juego deportivos.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. Deben estar inscritas las delegaciones dentro de la edición de los juegos deportivos.	
Entradas / Pasos de ejecución: Para conformar el medallero de una edición de juegos deportivos se mostrara las delegaciones participantes en dicha edición y se dará la opción de agregar o disminuir medallas a una delegación de el color deseado.	
Resultados esperados: Se conforma el medallero sin que se generen errores.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 108. Prueba 2 del módulo Emulación

Caso de prueba de aceptación	
Código: HU10_P2	Historia de Usuario: 10
Nombre: Conformar tabla de puntuación.	
Descripción: Prueba para la funcionalidad conformar la tabla de puntuación obtenida por cada una de las delegaciones.	
Condiciones de Ejecución: El usuario debe estar autenticado en el sistema. Se usará un usuario con datos válidos. Deben estar inscritas las delegaciones dentro de la edición de los juegos deportivos.	
Entradas / Pasos de ejecución: Para conformar la tabla de puntuación se mostrarán las delegaciones inscritas en la edición de los juegos deportivos en curso y se dará la opción de adicionar o restar puntos a una delegación	

determinada.
Resultados esperados: Se conforma la tabla de puntuación correctamente.
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 Interfacultades".

Conclusiones

Con la culminación del presente trabajo se ha llegado al término de un largo viaje. Aún cuando se es consciente de lo mucho que resta por aprender puede decirse que durante el transcurso del mismo los autores han adquirido un número de habilidades que dan acabado a su formación como ingenieros informáticos.

- Se elaboró un sistema informático completamente funcional para la gestión de la información de los Juegos Deportivos Interfacultades a tono con las exigencias de los clientes, titulado “Sistema para la Gestión de la Información de los Juegos Deportivos Interfacultades”.
- El sistema realizado solucionó una serie de problemas detectados inicialmente que atentaban contra la calidad de la organización y desarrollo de los juegos deportivos.
- Se desarrolló una aplicación extensible a diferentes eventos deportivos que se puedan desarrollar tanto en la UCI como en otros centros universitarios del país.
- Se comprobó lo beneficioso que resulta la incorporación de estrategias de desarrollo rápido (RAD) para este tipo de proyectos donde la combinación del framework symfony y la metodología XP potenciaron una implementación más efectiva de la aplicación.
- Los resultados arrojados en el transcurso de la investigación fueron recopilados en el presente documento el cual podrá consultarse en futuras investigaciones.

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.
- Realizar estudios más profundos con el fin de incorporar funcionalidades que permitan su uso en eventos deportivos de mayor envergadura que rebase los marcos del deporte universitario.
- Potenciar el uso en la facultad del framework symfony por las ventajas que brinda su utilización en aplicaciones de este tipo.
- Continuar el trabajo con el objetivo de incorporar al software nuevas funcionalidades.

Referencias Bibliográficas

1. **Quiroga, Lic. Lourdes Aja.** Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones. [En línea] http://bvs.sld.cu/revistas/aci/vol10_5_02/aci04502.htm
2. **Coubertin.** Memorias Olímpicas. [En línea] <http://www.tiroalcorcon.com/coubertin.htm>
3. Aplicación Web. [En línea] http://es.wikipedia.org/wiki/Aplicacion_web
4. Aplicaciones Web vs. Escritorio. [En línea] <http://alxplus.blogspot.com/2006/08/aplicaciones-web-vs.html>
5. HTML y XHTML. [En línea] http://www.librosweb.es/xhtml/capitulo1/html_y_xhtml.htm
6. CSS. [En línea] <http://www.librosweb.es/css/capitulo1.html>
7. PERL. [En línea] <http://www.desarrolloweb.com/articulos/541.php>
8. Desarrollo Web. [En línea] <http://www.maestrosdelweb.com/editorial/phpintro>
9. **Castillo, A.** Características de PHP. [En línea] http://www.programemos.com/index.php?option=com_content&task=view&id=279&Itemid=223
10. MySQL. [En línea] <http://es.wikipedia.org/wiki/MySQL>
11. Zope. [En línea] <http://es.wikipedia.org/wiki/Zope>
12. **Penadés, Patricio Letelier y M^a Carmen.** *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. España: Universidad Politécnica de Valencia : s.n., 2006.
13. symfony la guía definitiva. [En línea] http://www.librosweb.es/symfony_1_2
14. Aptana Studio. [En línea] <http://www.desarrolloweb.com/articulos/aptana-studio.html>
15. Viva PHP! Eclipse PDT 2.0. [En línea] <http://www.vivaphp.com.ar/soft/eclipse-pdt-2.0.html>
16. Versión traducida. . [En línea] <http://easyeclipse.org/>
17. Viva Linux! EasyEclipse: la manera fácil de instalar Eclipse. [En línea] <http://www.vivalinux.com.ar/soft/easy-eclipse>
18. **Joskowicz, Jose.** *Reglas y Prácticas en eXtreme Programming*. 2008.

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. A gentle introduction.2006 [citado enero 2009]; Disponible en:
<http://www.extremeprogramming.org>.
5. Qué es Javascript [citado enero 2009]; Disponible en:
<http://www.desarrolloweb.com/articulos/25.php>
6. Guía Breve de CSS [citado enero 2009]; Disponible en:
<http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>
7. **EGUÍLUZ, Javier.** Introducción a CSS. 2008 [citado enero 2009]; Disponible en:
<http://librosweb.es/css/>
8. Qué es Python [citado enero 2009]; Disponible en:
<http://www.desarrolloweb.com/articulos/1325.php>
9. Qué es Perl [citado enero 2009]; Disponible en: <http://www.desarrolloweb.com/articulos/541.php>
10. Desarrolloweb.com. "Programación en PHP" [citado enero 2009]; Disponible en:
<http://www.desarrolloweb.com/manuales/>
11. Maestros del Web Los Frameworks de PHP [citado febrero 2009]; Disponible en:
<http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo/>
12. Manual de PHP 5 [citado enero 2009]; Disponible en:
<http://www.desarrolloweb.com/manuales/58/>
13. Sitio oficial de PHP [citado enero 2009]; Disponible en: <http://www.php.net/>
14. Desarrolloweb.com Manual Ajax práctico - Taller Ajax [citado enero 2009]; Disponible en:
<http://www.desarrolloweb.com/manuales>
15. **EGUÍLUZ, Javier.** Introducción a AJAX. [citado enero 2009] Disponible en: <http://librosweb.es/ajax/>
16. **Diego Samir Melo Solarte** Base datos con software libre [citado enero 2009]; Disponible en:
www.umanizales.edu.co/programs/ingenieria/ventana/ventana12/articulo25.pdf
17. **KABIR, M. J.** La biblia de Servidor Apache 2, 2003, Vol: I.
18. **Fabien Potencier, François Zaninotto.** symfony la guía definitiva 2008 [citado febrero 2009];
Disponible en: http://www.librosweb.es/symfony_1_2

19. **Fabien Potencier**. El Tutorial Jobeet 2009 [citado febrero2009]; Disponible en:
<http://www.librosweb.es/jobeeet/>
20. Sitio Web de symfony [citado febrero 2009]; Disponible en: <http://www.symfony-project.org/>
21. Lista de correos de usuarios de symfony [citado febrero 2009]; Disponible en:
<http://groups.google.es/group/symfony-es>
22. El wiki de symfony [citado febrero 2009]; Disponible en: <http://trac.symfony-project.org/>

Glosario

Releases: Nueva versión de una aplicación informática.

IDE: Un entorno de desarrollo integrado o, en inglés, *Integrated Development Environment*, es un programa compuesto por un conjunto de herramientas para un programador.

Plug-in: “Parche” para un programa que le añade características nuevas.

PDT: PHP Development Tools, entorno de desarrollo PHP para Eclipse.

UML: Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

Frameworks: Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software, para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

MVC: Modelo Vista Controlador. Patrón de arquitectura muy utilizado en la creación de aplicaciones web.

HTTP: HyperText Transfer Protocol (Protocolo de transferencia de hipertexto). Es el protocolo usado para intercambiar archivos (texto, gráfica, imágenes, sonido, video y otros archivos multimedia) en la World Wide Web.

DOM: El *Document Object Model* (una traducción al español no literal, pero apropiada, podría ser *Modelo en Objetos para la representación de Documentos*), abreviado DOM, es esencialmente una interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos.

XMLHttpRequest: También referida como XMLHTTP (Extensible Markup Language / Hypertext Transfer Protocol), es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores WEB. Para los datos transferidos se usa cualquier codificación basada en texto, incluyendo: texto

plano, XML, JSON, HTML y codificaciones particulares específicas. La interfaz se presenta como una clase de la que una aplicación cliente puede generar tantas instancias como necesite para manejar el diálogo con el servidor.

XML: Siglas en inglés de *Extensible Markup Language* («lenguaje de marcas»), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos.

XSLT: XSLT o Transformaciones XSL es un estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML. Las hojas de estilo XSLT - aunque el término de hojas de estilo no se aplica sobre la función directa del XSLT - realizan la transformación del documento utilizando una o varias reglas de plantilla. Estas reglas de plantilla unidas al documento fuente a transformar alimentan un procesador de XSLT, el que realiza las transformaciones deseadas poniendo el resultado en un archivo de salida, o, como en el caso de una página web, las hace directamente en un dispositivo de presentación tal como el monitor del usuario.

JSON: Acrónimo de "JavaScript Object Notation", es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

ORM: El mapeo objeto-relacional (más conocido por su nombre en inglés, Object-Relational mapping,) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.