

Universidad de las Ciencias Informáticas

Facultad 1



**Aplicación web para la gestión de los procesos
extensionistas de la Facultad 1**

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

Autores: Lidia del Carmen Leyva Feijó


Alejandro López Rodríguez

Tutores: Ing. Geidis Sánchez Michel

Ing. Yareidis Barcena Calzado

La Habana, 24 de junio del 2015

“Año 57 de la Revolución”



“La inteligencia consiste no sólo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica”.

Aristóteles

Declaración de autoría:

Declaramos que somos los únicos autores del trabajo “Aplicación web para la gestión de los procesos extensionistas de la Facultad 1”, y autorizamos a la Facultad 1, de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Autores:

Lidia del Carmen Leyva Feijoó

Alejandro López Rodríguez

Tutores:

Ing. Geidis Sánchez Michel

Ing. Yarileidis Barcena Calzado

Nombre y apellidos de la tutora: Ing. Geidis Sánchez Michel.

Profesora asistente de la Universidad de las Ciencias Informáticas.

E-mail: gsanchez@uci.cu.

Ingeniero en Ciencias Informática 2007.

Función: Vicedecana de extensión y residencia de la Facultad 1.

Nombre y apellidos de la tutora: Ing. Yarileidis Barcena Calzado.

Profesora instructora de la Universidad de las Ciencias Informáticas.

E-mail: ybarcena@uci.cu.

Ingeniero en Ciencias Informática 2008.

Función: Profesora principal de tercer año.



Agradecimientos:

Alejandro

- ❖ *A mi mamá, y hermanos sobre todo Adriel porque son las personas más importante en mi vida y sin ellos este día solo hubiese ocurrido en sueños.*
- ❖ *A mi papá y Dévora por ser comprensivos y ayudarme en todo lo que han podido.*
- ❖ *A mis tutoras y profesores en toda la carrera especialmete para Geidis que ha sido como una madre.*
- ❖ *A mi gran amigo Carlos Cecilio, a mi amiga y compañera de tesis Lidia, a Lisandra, Yanet, Roberto, Gretel, Kirenia, Xiuny, Laura, Lázaro, Victor y a mi gente de Ciego.*
- ❖ *A Yoel por haberme ayudado durante estos últimos años, y por enseñarme que la vida es una sola y hay que aprender de ella.*
- ❖ *Gracias a Cuba y la Revolución por tenerme como su hijo y hacerme partícipe de esta misión que es mantener una patria libre, informatizada, nuestra.*
A todos gracias...

Lidia

- ❖ *A mi mayor tesoro, mi hija por ser la bendición más linda que me ha dado la vida y porque sin ella no hubiese alcanzado este sueño.*
- ❖ *A mi mamá, Sonia, mis abuelitos y familia por apoyarme siempre en todo momento, darme todo su amor y enseñarme a luchar por mis sueños.*
- ❖ *A mis tutoras en especial a Geidis por acogerme como una más de su equipo, pero en especial por ser más que una tutora, una amiga.*
- ❖ *A todos los profesores de la carrera que más que impartirme materias me educaron para la vida en especial a Bárbara Triana (Tata) y Madelín Haro.*
- ❖ *A todos mis amigos, por ser tan especiales para mí y reiterarme la frase "Quién tiene amigos, tiene un pueblo", en especial a Santiago, Jorge Luis, Alejandro mi fiel amigo y compañero de tesis, Julia, Kirenia, Vismar, Mónica, Isef, Laura, Xiuny, Marelis, Ernestina, Daniela, Daniel y mil gracias Roberto por formar parte de este sueño.*
- ❖ *Gracias a Cuba, Fidel y a la Revolución por brindarme la oportunidad de formar parte de una patria libre, informatizada y nuestra.*
- ❖ *Gracias Dios...*

Alejandro

- ❖ *A la persona que más amo, mi madre.*

Lidia

- ❖ *A la persona más importante de mi vida, mi hija por ser la luz de mis días.*

Resumen:

En la actualidad existe una tendencia de que la información sea manejada mediante sistemas computarizados facilitando de esta forma la interacción con la misma y agilizar el proceso de la toma de decisiones. En la Universidad de las Ciencias Informáticas (UCI), específicamente en el Vicedecanato de Extensión y Residencia de la Facultad 1, se desarrollan varios procesos extensionistas, entre los que se encuentran: la Reunión de Estudios Militares (REM), el Trabajo Socialmente Útil (TSU), el Festival de Artistas Aficionados (FAA), los Juegos Deportivos, así como el proceso evaluativo de la residencia estudiantil. La gestión de la información derivada de la realización de cada uno de estos eventos es llevada a cabo de forma manual, trayendo consigo una mayor inversión de tiempo y además el riesgo de que se introduzca información duplicada y errores. Por esta razón la presente investigación propone la realización de una aplicación web que permite la gestión de la información. El documento recoge los resultados de la investigación, describiéndose las principales características de los sistemas analizados. En el mismo se explica la arquitectura y el diseño de la solución propuesta, se describen las herramientas y tecnologías que se utilizan, así como los artefactos que se generan en el proceso de desarrollo guiado por la metodología XP.

Palabras clave: aplicación web, gestión de información, proceso extensionista.

Índice

Introducción1

Capítulo 1. Fundamentación teórica.....5

 1.1 Introducción 5

 1.2 Conceptos asociados al problema5

 1.3 Análisis de soluciones existentes..... 8

 1.4 Ambiente de desarrollo 11

Capítulo 2. Análisis y diseño del sistema 19

 2.1 Introducción 19

 2.2 Características del sistema 19

 2.3 Propuesta de solución..... 20

 2.4 Patrones 33

 2.5 Diseño del sistema 35

 2.6 Diseño de la base de datos..... 35

Capítulo 3. Implementación y pruebas del sistema.....37

 3.1 Introducción 37

 3.2 Patrones de diseño..... 37

 3.3 Implementación del sistema..... 41

 3.4 Estándares de codificación 42

 3.5 Pruebas 43

 3.6 Diagrama de despliegue 51

Conclusiones generales..... 52

Recomendaciones 53

Referencias bibliográficas 54

Anexos..... 59

Figura 1: Modelo Vista Controlador	34
Figura 2: Módulo Gestionar FAA.....	36
Figura 3: Patrón <i>Singleton</i>	38
Figura 4: Patrón Controlador.	39
Figura 5: Patrón Alta cohesión.	39
Figura 6: Patrón Bajo acoplamiento.	40
Figura 7: Patrón creador.....	40
Figura 8: Patrón mediador.....	41
Figura 9: Nomenclatura de las clases.	42
Figura 10: Nomenclatura de las funcionalidades.	43
Figura 11: Nomenclatura de las variables.	43
Figura 12: Resultados de las pruebas.....	50
Figura 13: Diagrama de despliegue.....	51
Figura 14: Módulo Gestionar personal.....	73
Figura 15: Autenticación	74
Figura 16: Módulo Gestionar juegos deportivos	75
Figura 17: Módulo Gestionar evaluación en la residencia	75
Figura 18: Módulo Gestionar TSU	76
Figura 19: Módulo Gestionar REM.....	76

Tabla 1. Personal relacionado con el sistema	21
Tabla 2. HU_Gestionar evento artístico	27
Tabla 3. Prioridad de las historias de usuario.	28
Tabla 4. Estimación del esfuerzo por historia de usuario.....	30
Tabla 5. Primera iteración	31
Tabla 6. Segunda iteración	31
Tabla 7. Tercera iteración.....	32
Tabla 8. Cronograma de liberación.....	33
Tabla 9. CRC Gestionar FAA clase: evento_faa	35
Tabla 10. Tarea: Diseñar la vista de gestionar evento artístico.	41
Tabla 11. Prueba estructural de caja blanca para la funcionalidad: existe evento FAA.	44
Tabla 12. PA_Gestionar evento artístico	45
Tabla 13. PF_Acceso al sistema.....	46
Tabla 14. PS_Prueba de seguridad.....	48
Tabla 15. Datos asociados a prueba de carga y estrés.	49
Tabla 16. Registro de no conformidades.....	50
Tabla 17. HU_Gestionar obra.....	61
Tabla 18. HU_Gestionar actividades colaterales FAA.....	62
Tabla 19. HU_Generar reportes FAA.....	62
Tabla 20. HU_Autenticar usuario	62
Tabla 21. HU_Gestionar personal.....	63
Tabla 22. HU_Gestionar estructura deportiva.....	63
Tabla 23. HU_Gestionar evento deportivo.	63
Tabla 24. HU_Gestionar agrupación deportiva.....	64
Tabla 25. HU Gestionar actividades colaterales JD.....	64
Tabla 26. HU_Generar reportes JD.....	64
Tabla 27. HU_Gestionar cuartelería.....	65
Tabla 28. HU_Gestionar evaluación mensual en la residencia.....	65
Tabla 29. HU_Generar reportes residencia.....	65
Tabla 30. HU_Gestionar áreas TSU.....	65
Tabla 31. HU_Gestionar eventos TSU.....	66
Tabla 32. HU_Gestionar grupos TSU.....	66
Tabla 33. HU_Generar reportes TSU.....	66
Tabla 34. HU_Gestionar estructura de mando.	67
Tabla 35. HU_Gestionar eventos REM.....	67
Tabla 36. HU_Gestionar grupos REM.....	67
Tabla 37. HU_Generar reportes REM.....	68

Tabla 38. CRC Gestionar FAA clase: obra.....	68
Tabla 39. CRC Gestionar FAA clase: actividad_colateral.....	68
Tabla 40. CRC Gestionar FAA clase: reporte_faa	69
Tabla 41. CRC Gestionar juegos deportivos clase: estructura_deportiva	69
Tabla 42. CRC Gestionar juegos deportivos clase: evento_jd	69
Tabla 43. CRC Gestionar juegos deportivos clase: agrupación_jd	70
Tabla 44. CRC Gestionar juegos deportivos clase: actividad_colateral	70
Tabla 45. CRC Gestionar juegos deportivos clase: reporte_jd	70
Tabla 46. CRC Gestionar evaluación de residencia clase: cuarteria	70
Tabla 47. CRC Gestionar evaluación de residencia clase: reporte	71
Tabla 48. CRC Gestionar TSU clase: estructura_area.....	71
Tabla 49. CRC Gestionar TSU clase: evento_tsu.....	71
Tabla 50. CRC Gestionar TSU clase: grupos_tsu.....	72
Tabla 51. CRC Gestionar TSU clase: registro_tsu.....	72
Tabla 52. CRC Gestionar TSU clase: reporte_tsu	72
Tabla 53. CRC Gestionar REM clase: estructura_mando.....	72
Tabla 54. CRC Gestionar REM clase: evento_rem.....	73
Tabla 55. Tarea: Diseñar base de datos (BD)	76
Tabla 56. Tarea: Mapear la BD.	77
Tabla 57. Tarea: Conectar gestionar evento artístico con la base de datos (BD).....	77
Tabla 58. Tarea: Diseñar la vista de gestionar obra artística.	77
Tabla 59. Tarea: Conectar Gestionar obra artística con BD.....	78
Tabla 60. Tarea: Diseñar la vista de Gestionar actividades colaterales FAA.	78
Tabla 61. Tarea: Conectar Gestionar actividades colaterales FAA con BD.	78
Tabla 62. Tarea: Diseñar Generar reportes FAA.	79
Tabla 63. Tarea: Conectar la vista de Generar reporte FAA con la BD.....	79
Tabla 64. Tarea: Diseñar la vista de Gestionar estructura deportiva.	79
Tabla 65. Tarea: Conectar la vista de Gestionar estructura deportiva con la BD.	80
Tabla 66. Tarea: Diseñar la vista de Gestionar actividades colaterales JD.....	80
Tabla 67. Tarea: Conectar la vista de Gestionar actividades colaterales JD con la BD.	80
Tabla 68. Tarea: Diseñar la vista de Gestionar eventos deportivos.....	81
Tabla 69. Tarea: Conectar la vista de Gestionar eventos deportivos con la BD.....	81
Tabla 70. Tarea: Diseñar la vista de Gestionar agrupación deportiva.	81
Tabla 71. Tarea: Conectar la vista de Gestionar agrupación deportiva con la BD.	82
Tabla 72. Tarea: Diseñar la vista de Generar reporte JD.	82
Tabla 73. Tarea: Conectar la vista de Generar reporte JD con la BD.	82
Tabla 74. Tarea: Diseñar la vista de Gestionar cuartería.	83

Tabla 75. Tarea: Conectar la vista de Gestionar cuarterería con BD.....	83
Tabla 76. Tarea: Diseñar la vista de Gestionar evaluación mensual en la residencia.....	83
Tabla 77. Tarea: Conectar la vista de Gestionar evaluación mensual en la residencia con la BD.....	84
Tabla 78. Tarea: Diseñar la vista de Generar reportes residencia.....	84
Tabla 79. Tarea: Conectar la vista de Generar reportes en la residencia con la BD.....	84
Tabla 80. Tarea: Diseñar la vista de Gestionar áreas TSU.....	85
Tabla 81. Tarea: Conectar la vista de Gestionar áreas TSU con la BD.....	85
Tabla 82. Tarea: Diseñar la vista de Gestionar eventos TSU.....	85
Tabla 83. Tarea: Conectar la vista de Gestionar eventos TSU con la BD.....	86
Tabla 84. Tarea: Diseñar la vista de Gestionar grupos TSU.....	86
Tabla 85. Tarea: Conectar la vista de Gestionar grupos TSU con la BD.....	86
Tabla 86. Tarea: Diseñar la vista de Generar reportes TSU.....	87
Tabla 87. Tarea: Conectar la vista de Generar reportes TSU.....	87
Tabla 88. Tarea: Diseñar la vista de Gestionar estructura de mando de la REM.....	87
Tabla 89. Tarea: Conectar la vista de Gestionar estructura de mando con la BD.....	88
Tabla 90. Tarea: Diseñar la vista de Gestionar eventos REM.....	88
Tabla 91. Tarea: Conectar la vista de Gestionar eventos REM con la BD.....	88
Tabla 92. Tarea: Diseñar la vista de Gestionar grupos REM.....	89
Tabla 93. Tarea: Conectar la vista de Gestionar grupos REM con la BD.....	89
Tabla 94. Tarea: Diseñar la vista de Generar reportes REM.....	89
Tabla 95. Tarea: Conectar la vista del reporte de la REM con la BD.....	90
Tabla 96. Tarea: Establecer arquitectura base.....	90
Tabla 97. Tarea: Establecer prototipo de diseño.....	90
Tabla 98. Tarea: Diseñar vista de Autenticar usuario.....	91
Tabla 99. Tarea: Conectar autenticación con la BD.....	91
Tabla 100. Tarea: Diseñar la vista de Gestionar personal.....	91
Tabla 101. Tarea: Conectar la vista de Gestionar personal con BD.....	92
Tabla 102. Tarea: Diseñar la vista para cargar los datos a través de un excel a la aplicación.....	92
Tabla 103. Tarea: Conectar la vista de cargar datos con la BD.....	92
Tabla 104. PA_Autenticar usuario.....	93
Tabla 105. PA_Gestionar personal.....	93
Tabla 106. PA_Gestionar obra.....	94
Tabla 107. PA_Gestionar actividades colaterales FAA.....	94
Tabla 108. PA_Generar reportes FAA.....	95
Tabla 109. PA_Gestionar estructura deportiva.....	95
Tabla 110. PA_Gestionar actividades colaterales JD.....	96
Tabla 111. PA_Gestionar evento deportivo.....	97

Tabla 112. PA_Gestionar agrupación deportiva.....	97
Tabla 113. PA_Generar reportes JD.	98
Tabla 114. PA_Gestionar cuarterería.	98
Tabla 115. PA_Gestionar evaluación mensual en la residencia.....	99
Tabla 116. PA_Generar reportes residencia.....	100
Tabla 117. PA_Gestionar áreas TSU.....	100
Tabla 118. PA_Gestionar eventos TSU.....	101
Tabla 119. PA_Gestionar grupos TSU.....	101
Tabla 120. PA_Generar reportes TSU.....	102
Tabla 121. PA_Gestionar estructura de mando.	102
Tabla 122. PA_Gestionar eventos REM.....	103
Tabla 123. PA_Gestionar grupos REM.....	104
Tabla 124. PA_Gestionar grupos REM.....	104
Tabla 125. Prueba estructural de caja blanca para la funcionalidad: obtener cantidad de evento FAA.	105
Tabla 126. Encuesta realizada_01.....	105
Tabla 127. Encuesta realizada_02.....	106
Tabla 128. Encuesta realizada_03.....	106

Introducción

La aparición y uso de las Tecnologías de la Información y las Comunicaciones (TIC) ha sido un salto cualitativo en el desarrollo científico técnico a escala mundial. Su utilización supone importantes logros en múltiples esferas de la sociedad como la economía, la educación, la salud y en el desarrollo social, proporcionando recursos y ahorro de tiempo, al simplificar y agilizar los procesos de gestión, así como la toma de decisiones.

Cada día el uso de las computadoras y medios tecnológicos se hace más necesario. El sistema de educación cubano trata de aprovechar las facilidades que ofrecen las mismas, aspirando informatizar la mayor cantidad de áreas posible y de este modo mejorar la calidad y el control del trabajo mediante el desarrollo de software y herramientas tecnológicas.

Cuba a pesar de ser un país bloqueado, trabaja actualmente en pos de lograr una digitalización de la mayor cantidad de información y procesos, aplicándose en diferentes esferas de la sociedad, con este propósito surge la Universidad de las Ciencias Informáticas (UCI) en el año 2002, con el fin de llevar la informatización a diferentes regiones del país a partir del desarrollo de software y sistemas propios (1).

Actualmente, en el Sistema de Educación Superior Cubano se concibe y potencia la función de la extensión universitaria como una de las principales vertientes de trabajo, apreciándose como el elemento más dinámico e integrador del vínculo universidad-sociedad (2).

De manera particular, la UCI ha atendido y perfeccionado los procesos que integran y garantizan su funcionalidad y eficiencia, resolviendo gradualmente sus carencias en las diferentes áreas a partir del desarrollo de aplicaciones, entre estas, las que hacen posible el trabajo con la información, su organización, control y procesamiento.

En esta institución se aplica un modelo de formación que vincula cuatro procesos sustantivos: formación, investigación, producción y extensión. Este último contempla diferentes subprocesos tales como: Trabajo Socialmente Útil (TSU), Reunión de Estudios Militares (REM), Festivales de Artistas Aficionados (FAA), Juegos Deportivos, así como la atención a la Residencia Estudiantil. Todos estos procesos son atendidos por los Vicedecanatos de Extensión y Residencia de cada facultad. Actualmente en el Vicedecanato de

Extensión y Residencia de la Facultad 1, el control de los procesos extensionistas se realiza de forma manual y la información es almacenada haciendo uso del paquete ofimático Open Office inherente al sistema operativo NOVA, provocando en ocasiones que los datos se dupliquen o se pierdan. La información proviene de diversas fuentes, lo que dificulta el proceso de generar reportes para realizar el análisis oportuno de los diferentes eventos que se llevan a cabo en el área de extensión y residencia.

Teniendo en cuenta lo expuesto previamente se plantea como **problema a resolver** ¿Cómo contribuir a la gestión de los procesos extensionistas de la Facultad 1? Una vez identificado el problema, se hace necesario enfocar la investigación en los sistemas para la gestión de procesos, lo cual constituye el **objeto de estudio** de la misma; tomando como **campo de acción**, las aplicaciones web para la gestión de procesos extensionistas de la Facultad 1.

Para dar solución al problema planteado se define como **objetivo general**: desarrollar una aplicación web para la gestión de los procesos extensionistas de la Facultad 1.

A partir del objetivo general, se derivan los **objetivos específicos** que se enuncian a continuación:

1. Definir el marco teórico de la investigación.
2. Diseñar una aplicación web para la gestión de los procesos extensionistas desarrollados en la Facultad 1.
3. Implementar la aplicación para la gestión de los procesos extensionistas desarrollados en la Facultad 1.
4. Validar la aplicación web mediante pruebas de software.

Partiendo del objetivo general de la presente investigación, se define como **idea a defender**: si se desarrolla una aplicación web para la gestión de los procesos extensionistas de la Facultad 1, se logrará un mejor control y seguimiento de los procesos.

Para cumplir los objetivos se plantean las siguientes **tareas de la investigación**:

1. Definición de los principales conceptos asociados al objeto de la investigación, así como a su campo de acción, para un mejor entendimiento de los mismos.

2. Análisis de la información existente relacionada a los servicios web para la gestión de procesos.
3. Selección de las principales herramientas y tecnologías que se ajustan a las necesidades de la investigación para el desarrollo del sistema propuesto.
4. Identificación de las principales funcionalidades de la aplicación web.
5. Diseño de la aplicación web propuesta.
6. Creación de los artefactos para la fase de implementación.
7. Implementación de la aplicación web propuesta.
8. Ejecución de pruebas para la validación del correcto funcionamiento de la aplicación web.

En la investigación se emplearon diferentes métodos científicos:

Métodos teóricos:

El **método analítico-sintético**: se utilizó para sintetizar todas las citas, apuntes y datos tomados al respecto. La revisión bibliográfica se realizó sobre un conjunto de libros, publicaciones, monografías y documentos en soporte electrónico, que se encuentran situados en páginas web, trabajos de curso, entre otros.

El **método histórico-lógico**: se utilizó para el trabajo recopilatorio sobre el proceso de desarrollo de los sistemas de gestión, así como las aplicaciones web para la gestión de información y procesos.

El **método modelación**: permitió realizar las actividades encaminadas a la adaptación o construcción de modelos a través de los cuales se representan parcial o totalmente las cualidades esenciales del fenómeno objeto de estudio en el desarrollo de actividades (3).

Métodos empíricos:

El **método de observación**: se utilizó para obtener de forma directa, la información de la realidad objetiva del comportamiento de la gestión de los procesos extensionistas que se desarrollan en Vicedecanato de Extensión y Residencia de la Facultad 1 de la Universidad de las Ciencias Informáticas.

El **método entrevista**: se realizó para obtener la información referente a como se gestiona actualmente la información en el Vicedecanato de Extensión y Residencia de la Facultad 1 así como el nivel de aceptación por parte de todo el personal involucrado en el proceso.

Como **posibles resultados** se espera:

- Lograr una aplicación web que permita controlar y gestionar la información del Vicedecanato de Extensión y Residencia de la Facultad 1.

El documento está estructurado en tres capítulos, los cuales se describen a continuación:

- **Capítulo 1. Fundamentación teórica:** en este capítulo se estudian los conceptos fundamentales asociados al dominio del problema para un mejor entendimiento de la investigación. Incluye un estudio del estado del arte de sistemas homólogos a nivel nacional e internacional, así como la descripción de la metodología, tecnologías y herramientas empleadas en el desarrollo de la aplicación web.
- **Capítulo 2. Análisis y diseño del sistema:** en este capítulo se describen las características de la aplicación a desarrollar. Se realizan las especificaciones de las funcionalidades de la aplicación web y se elabora el plan de entregas y de iteraciones.
- **Capítulo 3. Implementación y pruebas del sistema:** en este capítulo se desarrolla la propuesta de solución para luego llevar a cabo la validación mediante las pruebas de software.

Capítulo 1. Fundamentación teórica

1.1 Introducción

En este capítulo se abordan diversos conceptos asociados al problema en cuestión que resultan útiles para un mejor entendimiento del mismo, se caracteriza el objeto de estudio, se realiza un estudio del estado del arte de sistemas homólogos existentes. Se establece la selección del ambiente de desarrollo para dar cumplimiento a la propuesta de solución.

1.2 Conceptos asociados al problema

En el siguiente epígrafe se abordarán los principales conceptos asociados al dominio del problema, haciendo énfasis en los conceptos de gestión, sistemas de gestión, procesos y gestión universitaria para lograr un mayor entendimiento del problema en cuestión.

1.2.1 Gestión

Se denomina gestión al correcto manejo de los recursos de los que dispone una determinada organización, por ejemplo: empresas, organismos públicos y organismos no gubernamentales (4).

La gestión se considera un **proceso** en el cual se identifican ciertas etapas. La primera de ellas es la **planificación**, es en esta etapa donde se fijarán los objetivos a corto y largo plazo y el modo en que serán alcanzados. Es a partir de esta organización donde se determinarán el resto de las etapas. Luego sigue la etapa de **organización**, momento en el cuál los gestores determinan detalladamente el procedimiento para alcanzar los objetivos formulados anteriormente. Para ello son creadas las relaciones de trabajo y quién las liderará. Dicho de otra manera, se crea la estructura que organizará la institución. La tercer etapa es la de **liderazgo**, en este caso se intenta que el personal posea una dirección y motivación, de tal manera que resulte posible alcanzar los objetivos. Por último debe ser mencionado el **control**, en este caso el o los gestores examinan si la planificación es respetada y los objetivos son cumplidos. Para ello deben ser capaces de realizar ciertas correcciones si las normas no son acatadas (5).

1.2.2 Sistemas de gestión

Los sistemas de gestión permiten aprovechar y desarrollar el potencial existente en una organización posibilitando el alcance de los objetivos de la misma mediante una serie de estrategias, como el enfoque centrado en la gestión y el pensamiento disciplinado, de forma tal que posibilita:

- Definir una política interna que fije metas u objetivos y el compromiso de lograr un alto nivel de desempeño, cumpliendo con todos los requisitos legales, encaminados a la mejora continua, tendiendo a proveer los recursos adecuados y apropiados para implementar la política trazada.
- Incorporar el concepto de “competencia” y se establece la necesidad de evaluar la efectividad y eficiencia del personal para las tareas asignadas.
- Inducir y fomentar la formación y capacitación del personal en su área.
- Concientizar al personal de la relevancia e importancia de sus actividades y de cómo esta contribuye a la consecución de los objetivos de la organización.
- Medir los objetivos trazados (6).

1.2.3 Software de gestión

Las aplicaciones o software de gestión son aquellas diseñadas para sustituir uno o varios procedimientos, tanto comerciales como administrativos, que habitualmente realiza una persona en una empresa o institución de forma presencial o por un software que le permite realizar al cliente los mismos procedimientos de forma no presencial, disminuyendo el esfuerzo empleado para la realización de los mismos.

1.2.4 Proceso

Un proceso es una secuencia de pasos dispuesta con algún tipo de lógica que se enfoca en lograr algún resultado específico. Los procesos son mecanismos de comportamiento que diseñan los hombres para mejorar la productividad de algo, para establecer un orden o eliminar algún tipo de problema (7).

Los procesos pueden ser clasificados en función de varios criterios, la clasificación más habitual de los procesos en la práctica es distinguir entre: estratégicos, clave o de apoyo (8).

- Los procesos clave son denominados operativos y son propios de la actividad de una empresa.
- Los procesos estratégicos son aquellos procesos mediante los cuales una empresa desarrolla sus estrategias y define sus objetivos.
- Los procesos de apoyo, o de soporte son los que proporcionan los medios (recursos) y el apoyo necesario para que los procesos clave se puedan llevar a cabo.

1.2.5 Gestión universitaria

El Master en Ciencias Jesús Cejas Montero plantea: la gestión universitaria está compuesta por un conjunto de factores (recursos, procesos y resultados) que deben estar al servicio y contribuir positivamente al desarrollo de la docencia, la investigación y la extensión, cuyo objetivo básico es conducir al desarrollo integral de la institución y no a una asociación de unidades académicas aisladas.

La universidad siempre ha estado y estará vinculada a la gestión universitaria, esta conforma la manera de dirigir y orientar la universidad y por ende el conocimiento. La mejor gestión universitaria permite la formación e investigación de manera organizada y con retroalimentación continua, basada en la innovación y los recursos humanos (9).

1.2.6 Fundamentos que definen el proceso de Extensión Universitaria

El Máster en Ciencias en proyectos educativos y sociales Julio Cedeño Ferrín expresa: la extensión universitaria constituye una de las funciones esenciales de la universidad y es síntesis del resto de sus funciones para el logro de su pertinencia social; por lo tanto, integra la docencia y la investigación. De ahí que ella sea un atributo inherente de la universidad, fuertemente vinculada con la sociedad que la alberga y sostiene.

El proceso de gestión extensionista es aquel que, como resultado de las relaciones sociales que se dan entre los sujetos que en él participan, está dirigido de un modo sistémico y eficiente, a la promoción de cultura para la comunidad universitaria (objetivo), con vistas a la solución del (problema) social; la necesidad de contribuir al desarrollo cultural de la comunidad, mediante la apropiación de la cultura que ha acumulado la sociedad en su desarrollo (contenido); a través de la participación activa de la comunidad universitaria y extrauniversitaria (método); planificada en el tiempo y observando ciertas estructuras organizativas (forma); con ayuda de ciertos objetos (medio); instrumentando indicadores que permitan

medir la calidad (evaluación) y cuyo movimiento está determinado por las relaciones causales entre sus componentes y de ellos con la sociedad (leyes) que constituyen su esencia (10).

1.2.7 ¿Qué es una aplicación web?

Una aplicación web es un documento electrónico que contiene información, cuyo formato se adapta para ser insertado en la *World Wide Web*, de manera que los usuarios a nivel mundial puedan acceder a la misma haciendo uso de un navegador, visualizándola con un dispositivo móvil como un smartphone o un monitor de computadora (11).

1.2.8 Características generales de una aplicación web

Una aplicación web que reside en uno o varios servidores normalmente está estructurada en tres capas: servidor web, lógica de la aplicación y un sistema gestor de bases de datos (en adelante SGBD). El primero de ellos mantiene la comunicación con el navegador del usuario que es quien proporciona la interfaz de acceso al servicio web. La lógica de la aplicación está formada por los scripts o programas que se ejecutan en el servidor para atender las solicitudes del usuario, y el SGBD donde se guardan los datos con los que trabaja el servicio. Las tres capas pueden estar montadas sobre una o varias máquinas dependiendo del volumen de carga que tenga que soportar el servicio, ejecutándose sobre algún sistema operativo (12).

Ventajas de las aplicaciones web

- No requieren actualizar procedimiento puesto que todas las funciones están implementadas en el servidor y enviados automáticamente a los usuarios.
- Se integran fácilmente en otros procedimientos web del lado del servidor, tales como el correo electrónico y la búsqueda de información.
- Proporcionan compatibilidad entre plataformas en la mayoría de los casos debido a que operan dentro de una ventana de un navegador web (13).

1.3 Análisis de soluciones existentes

En este epígrafe se analizarán los sistemas homólogos existentes a nivel nacional e internacional,

haciendo especial énfasis en los que se encuentran alojados en la propia Universidad, para identificar características que deben tenerse en cuenta en la solución que será propuesta.

1.3.1 Soluciones existentes a nivel internacional

AvaiBook Sports

Es un sistema que gestiona eventos deportivos de forma gratuita para que organizadores de eventos, empresas de cronometraje, clubes deportivos, circuitos de carreras, puedan mantener un control y gestión de las diferentes modalidades deportivas (14).

Permite gestionar cualquier evento deportivo, inscribirse de forma *online* en el evento, promocionar y difundir el evento, gestionar los premios, exportar el listado de los participantes inscritos en el evento en formato Excel, utilizando los filtros que proporciona. El sistema es totalmente compatible con cualquier sistema de cronometraje. Permite utilizar redes sociales como Facebook y Twitter para la difusión de los eventos.

Por sus características queda descartado su uso como solución al problema, teniendo en cuenta que la solución solo se enmarca en la gestión de eventos deportivos y no integra todos los procesos que se desean gestionar en el Vicedecanato de Extensión y Residencia de la Facultad 1.

Sistema de gestión patrimonial (SIGPAC)

La Sección de Patrimonio Cultural es una unidad especializada, orientada a gestionar políticas públicas e implementar actividades que promuevan la salvaguarda del patrimonio cultural inmaterial, apoyando principalmente el fortalecimiento de las manifestaciones y expresiones vivas de las culturas locales (15).

Este sistema permite gestionar los diferentes programas patrimoniales, promocionar y difundir los eventos que se desarrollan en el ámbito de la gestión del patrimonio cultural.

El sistema anterior queda descartado como solución al problema, teniendo en cuenta que solo se enmarca en la gestión patrimonial, no permite la creación de reportes y no integra todos los procesos que se desean gestionar en el Vicedecanato de Extensión y Residencia de la Facultad 1.

Tracktherace

Tracktherace facilita la creación de un sitio web para cada evento gestionado. Es una aplicación multiplataforma. El contenido de la misma es administrable por los gestores del evento; les permite ir colocando toda la información que se considere importante para los participantes y para el público en general, antes o después del evento: reglamento, raidbooks, resultados, enlaces a fotos o videos.

La plataforma soporta las inscripciones con registro en *Tracktherace* (con lo que los datos serán recordados para futuros eventos), o la inscripción sin registro. La inscripción puede tener información obligatoria u opcional. Los plazos de inscripción pueden ser configurados por los gestores para ser automáticamente abiertos o cerrados, evitando la interacción con el sistema en estos días concretos.

Queda descartado como solución al problema, teniendo en cuenta que solo se enmarca en la gestión de los eventos y no integra todos los procesos que se desean gestionar en el Vicedecanato de Extensión y Residencia de la Facultad 1.

1.3.2 Soluciones existentes a nivel nacional

CUBALITERARIA

La Editorial Electrónica Cubaliteraria, Portal del Instituto Cubano del Libro, es la ventana al mundo sobre el acontecer de la literatura cubana dentro y fuera de la Isla. Conforman su espacio informativo; ensayos, artículos noticiosos, columnas de autor, entrevistas, homenajes, convocatorias, entre otros perfiles.

Este portal está desarrollado en el lenguaje PHP (*Hypertext Pre-processor*), tiene diferentes secciones como son: Barquito de papel: (se le da promoción a algún libro de la feria y se deja publicada las promociones anteriores donde se hace una sintaxis del libro seleccionado), Inicio, Quienes Somos, de la Cuba Literaria, Autores, Directorios Web, Librerías, Baúl Literario, Boletín Electrónico, Publicaciones Literarias Digitales y una búsqueda avanzada por Título, Autor y Secciones (16).

Por sus características queda descartado su uso como solución al problema, teniendo en cuenta que se dedica solamente a la gestión de eventos culturales, la manera en que gestiona dichos eventos, no se ajusta a las necesidades existentes en el Vicedecanato de Extensión y Residencia de Facultad 1.

1.3.3 Soluciones existentes en la UCI

JDEPORTIVOS

En el año 2008, se crea el sitio JDEPORTIVOS con el objetivo de promover, impulsar y llevar el control estadístico de los eventos asociados a los Juegos Mella de la Universidad de las Ciencias Informáticas. Está elaborado sobre el Sistema de Gestión de Contenidos (*Content Management System*, CMS) Joomla, en la versión 1.0.11, actualmente carente de soporte, constituyendo una tecnología prácticamente obsoleta y además sobrescribe la información de un curso a otro lo que imposibilita llevar un historial de la información correspondiente a la realización de este evento.

El sistema descrito anteriormente queda descartado como posible solución al problema, teniendo en cuenta que no se especifica por cada deporte los atletas que obtuvieron resultados, se basa solamente en la gestión de eventos deportivos, lo cual indica que no sería un buen sistema para llevar a cabo el control de todos los procesos que se desean integrar en el Vicedecanato de Extensión y Residencia de Facultad 1.

El estudio de estos sistemas permitió determinar que actualmente no existe una aplicación web que logre integrar todos los procesos que se desarrollan en el Vicedecanato de Extensión y Residencia de la Facultad 1, por lo que se determina que resulta necesario la creación de una aplicación web que cumpla con la característica antes descrita. El análisis posibilitó la adquisición de buenas prácticas que influirán positivamente en la calidad de la solución propuesta.

1.4 Ambiente de desarrollo

Lograr un software de calidad implica la correcta selección de herramientas, metodologías y técnicas que permitan unificar la filosofía de trabajo. En esta sección se describe el ambiente de desarrollo que será utilizado en la propuesta de solución.

1.4.1 Servidor web

Un servidor web es un programa informático que procesa una aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente, generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente (17).

La clasificación según el tipo de prestaciones que brindan:

- **Servidor dedicado:** ofrece un único servicio telemático.
- **Servidor compartido:** ofrece varios servicios telemáticos.

Entre los servidores web más encontrados están: MICROSOFT_IIS, SUN_JAVA_SYSTEM_WEB SERVER, WAMP, XAMPP, APPSERV, APACHE, entre otros. Se seleccionó **APACHE** en su versión 2.2.22 para el desarrollo del sistema propuesto por las características descritas a continuación:

- Ofrece tecnología libre y de código abierto, otorgando transparencia y brindando la posibilidad de conocer lo que realmente se desea instalar.
- Trabaja en conjunto con gran cantidad de Lenguajes de Programación interpretados como PHP, Perl, soporte con CGI (*Common Gateway Interface*), Java, JSP (*Java Server Pages*) y otros lenguajes de script, el complemento ideal para los sitios web dinámicos que existen en la actualidad.
- Es un servidor web altamente configurable y de diseño modular, capaz de ampliar su funcionalidad y calidad de servicios (18).

1.4.2 Lenguajes de programación

Entre los lenguajes de programación existentes de código abierto para desarrollar aplicaciones orientadas a la web se encuentran dos grupos fundamentales de acuerdo con la arquitectura Cliente/Servidor: la programación del lado del servidor y la programación del lado del cliente. La programación del lado del cliente incluye aquellos lenguajes que son únicamente interpretados por una aplicación cliente como el navegador web, entre estos lenguajes se encuentran HTML, Java Script, Visual Basic y Java. Los lenguajes de programación del lado del servidor son los reconocidos e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él (19).

A continuación se describen los lenguajes de programación utilizados del lado del cliente:

HTML (*Hypertext Markup Language*)

HTML es un lenguaje sencillo desarrollado por el *World Wide Web Consortium*. Es un lenguaje estático fácil de aprender que permite preparar documentos web, se trata de marcas que controlan los distintos aspectos de la presentación y el comportamiento de sus elementos (20). Específicamente, HTML es el lenguaje con el que se escribe la estructura y la semántica del contenido de un documento web (21). Para el desarrollo del trabajo en cuestión se utilizó HTML en su versión 5.0.

JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (22). Para el desarrollo del trabajo en cuestión se utilizó el lenguaje JavaScript en su versión 1.8.

A continuación se describen los lenguajes de programación utilizados del lado del servidor:

PHP (*Hypertext Preprocessor*)

PHP es un lenguaje interpretado de propósito general ampliamente usado, está diseñado especialmente para el desarrollo web y puede ser incrustado dentro del código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX (y de ese tipo, como Linux o Mac OS X) y Windows (23).

Este lenguaje es gratuito, multiplataforma y rápido, con una gran librería de funciones y documentación, que lo hace útil para realizar varios tipos de aplicaciones web.

Las librerías de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red. Algunas de las más importantes capacidades de PHP son: compatibilidad con las bases de datos más comunes, como MySQL, mSQL, Oracle, Informix, y ODBC. Incluye funciones para el envío de correo electrónico, carga de archivos, crear dinámicamente en el servidor imágenes en formato GIF, animadas y

una gran lista de utilidades adicionales (24). Para el desarrollo de la solución propuesta se hizo uso del lenguaje PHP en su versión 5.5.9.

1.4.3 Administrador de Base de Datos pgAdmin III

PgAdmin III es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, es la más completa y popular con licencia *Open Source*. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, permitiendo su uso en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres.

PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. Soporta todas las características de PostgreSQL y facilita la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados y soporte para el motor de replicación Slony-I (25). Para llevar a cabo la realización del presente trabajo se utilizó el **pgAdmin III** en su versión 1.14.0.

1.4.4 Gestor de Base de Datos PostgreSQL

PostgreSQL es un SGBD relacional, orientado a objetos y libre, que tiene prestaciones y funcionalidades equivalentes a muchos gestores de bases de datos comerciales. Dentro de sus principales ventajas se encuentran:

- Soporta distintos tipos de datos.
- Posee una gran escalabilidad, haciéndolo idóneo para su uso en sitios web que atienden un gran número de solicitudes.
- Es extensible a través del código fuente y disponible sin costos adicionales.
- Soporte para los lenguajes más populares del medio: PHP, C++, Perl, Python, entre otros.
- Extensiones para alta disponibilidad, nuevos tipos de índices, datos especiales, minería de datos, entre otros.

- Es multiplataforma, disponible en Linux, Unix, Mac Os X y Windows, entre otros (26).

Se seleccionó PostgreSQL en su versión 9.4, debido a que es multiplataforma, confiable, estable, posee gran escalabilidad, control de concurrencia y funcionalidades que lo destacan como uno de los SGBD más potentes en la actualidad.

1.4.5 Metodología de desarrollo de software

XP (*Extreme Programing*)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define especialmente adecuada para proyectos con requisitos imprecisos, cambiantes y donde existe un alto riesgo técnico (27).

Programación Extrema o XP

- Metodología liviana de desarrollo de software.
- Conjunto de prácticas y reglas empleadas para desarrollar software.
- Basada en diferentes ideas acerca de cómo enfrentar ambientes muy cambiantes.

Un proyecto basado en la metodología XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos (28):

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

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

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

La metodología se basa en (29):

- **Pruebas Unitarias:** se basa en las pruebas realizadas a los principales procesos obteniéndose de esta forma los errores que pudieran ocurrir.
- **Refabricación:** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- **Programación en pares:** una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

Metodología a utilizar

Para llevar a cabo la realización del presente trabajo se determina usar la Metodología XP (*Extreme Programming*) teniendo en cuenta las características antes descritas, además de que esta metodología es muy adaptable a las necesidades requeridas, permitiendo analizar desde el punto de vista del cliente las principales funcionalidades a implementar en el desarrollo del producto.

1.4.6 Lenguaje unificado de modelado (UML)

UML es un lenguaje muy conocido y utilizado en el modelado de sistemas de software, permite visualizar, especificar, construir y documentar el sistema a desarrollar. Ofrece un estándar para detallar un plano del sistema (modelo), el cual incluye características conceptuales como procesos de negocio, funciones del sistema, expresiones de lenguaje de programación, esquemas de bases de datos, entre otros (30).

Debido a las características anteriormente mencionadas, se escoge UML en su versión 2.0 como lenguaje de modelado.

1.4.7 Herramienta CASE (*Computer Aided Software Engineering*)

Las herramientas CASE surgen para auxiliar a los desarrolladores de software, lo que permite el apoyo computarizado en todo o en parte del ciclo de vida del desarrollo de un sistema de software. Las herramientas CASE han surgido para dar solución a varios problemas inherentes al diseño del software, principalmente para la mejora de la calidad del desarrollo de sistemas de mediano y gran tamaño, y en segundo término, por el aumento de la productividad (31).

Visual Paradigm

Es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de software, análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El lenguaje de modelado UML ayuda a una rápida construcción de aplicaciones de calidad. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta CASE también proporciona abundantes tutoriales, demostraciones interactivas y proyectos UML (32). Para el desarrollo de la solución propuesta se utilizó la herramienta Visual Paradigm en su versión 8.0.

1.4.8 Entorno de desarrollo integrado NetBeans

NetBeans constituye una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero se utiliza para desarrollar en varios lenguajes de programación(33).

Características:

- Es un Entorno de Desarrollo Integrado (IDE) de código abierto y gratuito.
- Es multiplataforma, haciendo posible su uso en diversos sistemas operativos.
- Brinda las herramientas necesarias para el desarrollo en lenguajes como Java, XML, HTML, PHP.
- Presenta aplicaciones para la detección y tratamiento de errores.
- Tiene una amplia comunidad de desarrollo y soporte.

En el desarrollo del presente trabajo se hace uso del entorno de desarrollo integrado NetBeans IDE en su versión 8.0.

1.4.9 Framework de desarrollo

CodeIgniter

CodeIgniter es un framework para el desarrollo de aplicaciones, una herramienta para quienes crean páginas web usando PHP. Su meta es permitir desarrollar proyectos mucho más rápido que escribir el código desde cero, proporcionando una gran variedad de librerías para las tareas más comunes, así como una interfaz simple y una estructura lógica para acceder a estas librerías.

Por otra parte, CodeIgniter es justamente restrictivo sobre los caracteres en las cadenas URI para ayudar a minimizar las posibilidades de ataques utilizando esta vía. Construye de forma sencilla un conjunto de restricciones, validaciones y filtros. Implementa el patrón arquitectónico llamado *Modelo Vista Controlador* (MVC). Además contiene ayudas para la creación de aplicaciones PHP avanzadas y define una arquitectura de desarrollo que permite programar de una manera más ordenada a la vez que contiene diversas herramientas que ayudan a realizar aplicaciones más versátiles y seguras (34). Para el desarrollo del trabajo en cuestión se llevó a cabo el empleo del framework de desarrollo CodeIgniter en su versión 1.7.2.

JQuery

JQuery es una biblioteca gratuita de JavaScript, cuyo objetivo principal es simplificar las tareas de creación de páginas web responsivas, acorde a lo estipulado en la Web 2.0, la cual funciona en todos los navegadores modernos. Otra de las grandes ventajas de JQuery es que se enfoca en simplificar los scripts y en acceder o modificar el contenido de una página web (35). Para apoyar el desarrollo de la solución propuesta se utilizó el framework de desarrollo JQuery en su versión 1.3.2.

Capítulo 2. Análisis y diseño del sistema

2.1 Introducción

En este capítulo se realiza una descripción general de la propuesta de solución, definiendo los requisitos funcionales y no funcionales del sistema. Las funcionalidades se describen mediante las historias de usuario (HU). Se lleva a cabo la realización del plan de entrega, donde se indican las historias de usuario que se crearán para cada versión de la aplicación propuesta, así como las fechas en las que se publicarán dichas versiones. Se realiza el plan de iteraciones donde se muestran las HU a realizar en cada iteración según su prioridad en el negocio.

2.2 Características del sistema

El sistema propuesto consiste en una aplicación web, la cual tiene como objetivo fundamental la gestión de la información derivada de los procesos extensionistas de la Facultad 1. Dentro de las características del sistema propuesto, se encuentran, como lenguaje de programación PHP 5.5.9 del lado del servidor y JavaScript 1.8 como lenguaje de programación del lado del cliente, además se utiliza la tecnología AJAX para la comunicación asíncrona desde el navegador con el servidor web Apache en su versión 2.2.22. Se utilizó la versión 8.0 del Entorno de Desarrollo Integrado (IDE por sus siglas en inglés) NetBeans y como Sistema Gestor de Bases de Datos (SGBD) el PostgreSQL en su versión 9.4. Todo el desarrollo de la propuesta de solución es guiado a través de un proceso de desarrollo de software con enfoque ágil, en el que se aplican buenas prácticas de los procesos ágiles pues se cuenta con un equipo de desarrollo pequeño, el cliente forma parte de la organización y la rápida entrega de resultados tangibles es quien guía el desarrollo. El sistema contará con varios módulos, a través de los cuales, los usuarios podrán acceder, y si su rol lo permite, modificar las siguientes informaciones:

- **Autenticar usuario:** se encarga de la seguridad del sistema, recogiendo los datos de usuario y contraseña, si la persona que introduce los datos no es un usuario “conocido” o válido, no podrá tener acceso al sistema. En esta funcionalidad se establecen los permisos para cada rol de la aplicación.
- **Gestionar personal:** maneja la información relacionada con el personal que accede o forma parte de la aplicación.

- **Festival de artistas aficionados (FAA):** maneja información sobre los diferentes eventos artísticos pertenecientes al festival de artistas aficionados, donde se gestiona la información referente a las obras que forman parte del mismo, así como la asignación de medallas, permitiendo generar reportes estadísticos y generales del festival.
- **Evaluaciones en la residencia:** maneja información relacionada con la cuarterería, permitiendo obtener, modificar, actualizar, eliminar e insertar la misma, así como la asignación de esta a cada estudiante y el estado de la evaluación correspondiente. Permite asignar una evaluación mensual al estudiante en el área de la residencia y generar reportes con las informaciones antes descrita.
- **Juegos deportivos (JD):** maneja la información de la estructura deportiva y la relación persona-deporte, permitiendo insertar, obtener, modificar, actualizar y eliminar cada una de las estructuras deportivas, así como la gestión de las diferentes actividades colaterales y el personal involucrado en cada una de estas. Se obtienen reportes estadísticos y generales que muestran los resultados de los juegos deportivos.
- **Trabajo socialmente útil (TSU):** permite la gestión de las diferentes áreas del TSU, así como insertar, obtener, modificar, actualizar y eliminar los eventos y grupos del mismo. Permite asignar a cada estudiante a una respectiva área y la evaluación diaria correspondiente. Genera un reporte con la información de los estudiantes mostrando la participación de cada uno de ellos y los resultados obtenidos durante el TSU.
- **Reunión de estudios militares (REM):** maneja la información de la estructura de mando de la REM, composición de la compañía, pelotones y la participación diaria del estudiante, permitiendo dar un resultado final del cumplimiento de esta actividad. Genera un reporte con la información de cada uno de los estudiantes que cumplieron o quedaron pendientes de la REM.

2.3 Propuesta de solución

En la primera fase de la metodología XP, los clientes plantean a grandes rasgos las funcionalidades que son de interés para la elaboración del producto, transformándose en historias de usuario. Partiendo de la información obtenida, el equipo de desarrollo evaluará de forma general el tiempo de codificación, se familiarizará con las herramientas, tecnologías y prácticas que serán utilizadas en el proyecto, además, se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo para ello.

La fase de exploración toma de pocas semanas a pocos meses, dependiendo de la habilidad que tengan los programadores con las tecnologías seleccionadas. Es válido aclarar que las estimaciones realizadas en esta etapa son primarias, pues estas se basan en datos de alto nivel los cuales pueden variar a medida que se analicen con mayor cuidado y detalle en las siguientes fase (36).

2.3.1 Personal relacionado con el sistema

Uno de los principales factores que se deben tener en cuenta cuando se comienza el desarrollo de un sistema informático es la delimitación de la audiencia a la cual va dirigido el mismo, lo cual en algunas ocasiones, puede referirse a un personal relacionado con el sistema o alguien completamente ajeno a él. Se debe especificar que esta audiencia a su vez puede ser dividida en grupos atendiendo a sus necesidades. Se define como persona relacionada con el sistema, a toda aquella que de una manera u otra interactúe con este, obteniendo un resultado de uno o varios procesos que se ejecutan en el mismo. También son considerados como personas relacionadas con el sistema, aquellas que se encuentran involucradas en dichos procesos, participan en ellos, pero no obtienen ningún resultado de valor.

En la tabla 1 se muestra la relación entre los actores y la aplicación web.

Tabla 1. Personal relacionado con el sistema

Personal	Descripción
Vicedecana de Extensión y Residencia de la Facultad 1	Tiene completo acceso al sistema y posee el rol de Administrador.
Profesor de la Facultad 1	Tiene acceso al sistema solo para consultar la información, excepto aquellos profesores que se le asignan determinados permisos para desempeñar otras funciones.
Instructor de la Residencia 2	Tiene acceso solo al módulo de residencia, donde puede visualizar y modificar la información.
Estudiante de la Facultad 1	Tiene acceso al sistema donde solo pueden visualizar la información correspondiente a su participación en los diferentes eventos que se desarrollan en el Vicedecanato de Extensión y Residencia de la Facultad 1.
Invitado	Tiene acceso al sistema donde puede visualizar los reportes asociados a cada uno de los módulos de la aplicación.

A continuación se describe la lista de reserva del producto, compuesta por los requisitos funcionales y no funcionales de la aplicación web.

2.3.2 Requisitos funcionales de la aplicación web

1. Autenticar usuario

- 1.1. Iniciar sesión
- 1.2. Cerrar sesión

2. Gestionar personal

- 2.1. Insertar personal
- 2.2. Actualizar estado de un personal
- 2.3. Obtener lista de personal
- 2.4. Eliminar personal
- 2.5. Cargar Excel

3. Gestionar festival de artistas aficionados

- 3.1. Gestionar evento artístico
 - 3.1.1. Insertar evento artístico
 - 3.1.2. Obtener evento artístico
 - 3.1.3. Actualizar evento artístico
 - 3.1.4. Eliminar evento artístico
- 3.2. Gestionar obra
 - 3.2.1. Insertar obra
 - 3.2.2. Obtener obra
 - 3.2.3. Actualizar obra
 - 3.2.4. Eliminar obra
 - 3.2.5. Asignar persona a obra
 - 3.2.6. Asignar medalla a obra
- 3.3. Gestionar actividades colaterales FAA
 - 3.3.1. Insertar actividades colaterales
 - 3.3.2. Obtener actividades colaterales
 - 3.3.3. Actualizar actividades colaterales
 - 3.3.4. Eliminar actividades colaterales
 - 3.3.5. Asignar persona a actividades colaterales
- 3.4. Generar reportes FAA

4. Gestionar juegos deportivos

- 4.1. Gestionar estructura deportiva
 - 4.1.1. Insertar estructura deportiva
 - 4.1.2. Obtener estructura deportiva
 - 4.1.3. Actualizar estructura deportiva
 - 4.1.4. Eliminar estructura deportiva
- 4.2. Gestionar actividades colaterales JD
 - 4.2.1. Insertar actividades colaterales
 - 4.2.2. Obtener actividades colaterales
 - 4.2.3. Actualizar actividades colaterales
 - 4.2.4. Eliminar actividades colaterales
 - 4.2.5. Asignar persona a actividades colaterales
- 4.3. Gestionar evento deportivo
 - 4.3.1. Insertar evento deportivo
 - 4.3.2. Obtener evento deportivo
 - 4.3.3. Actualizar evento deportivo
 - 4.3.4. Eliminar evento deportivo
- 4.4. Gestionar agrupación deportiva
 - 4.4.1. Insertar agrupación deportiva
 - 4.4.2. Obtener agrupación deportiva
 - 4.4.3. Actualizar agrupación deportiva
 - 4.4.4. Eliminar agrupación deportiva
 - 4.4.5. Asignar persona a agrupación deportiva
 - 4.4.6. Asignar medalla a agrupación deportiva
- 4.5. Generar reportes JD
 - 4.5.1. Generar medallero por estudiante
 - 4.5.2. Generar medallero por agrupación deportiva

5. Gestionar evaluación de la residencia

- 5.1. Gestionar cuartería
 - 5.1.1. Insertar cuartería

- 5.1.2. Obtener cuartelería
- 5.1.3. Actualizar cuartelería
- 5.1.4. Eliminar cuartelería
- 5.1.5. Asignar persona a cuartelería
- 5.1.6. Asignar estado de evaluación a la cuartelería

5.2. Gestionar evaluación mensual en la residencia

- 5.2.1. Insertar evaluación
- 5.2.2. Obtener evaluación
- 5.2.3. Actualizar evaluación
- 5.2.4. Eliminar evaluación

5.3. Generar reportes de residencia

- 5.3.1. Generar reportes de la evaluación de la cuartelería
- 5.3.2. Generar reportes de evaluación mensual

6. Gestionar trabajo socialmente útil

6.1. Gestionar áreas TSU

- 6.1.1. Insertar áreas
- 6.1.2. Obtener áreas
- 6.1.3. Actualizar áreas
- 6.1.4. Eliminar áreas

6.2. Gestionar eventos TSU

- 6.2.1. Insertar eventos TSU
- 6.2.2. Obtener eventos TSU
- 6.2.3. Actualizar eventos TSU
- 6.2.4. Eliminar eventos TSU

6.3. Gestionar grupos TSU

- 6.3.1. Insertar grupos TSU
- 6.3.2. Obtener grupos TSU
- 6.3.3. Actualizar grupos TSU
- 6.3.4. Eliminar grupos TSU

6.4. Generar reportes TSU

7. Gestionar reunión de estudios militares

7.1. Gestionar estructura de mando

- 7.1.1. Insertar estructura de mando
- 7.1.2. Obtener estructura de mando
- 7.1.3. Actualizar estructura de mando
- 7.1.4. Eliminar estructura de mando

7.2. Gestionar eventos REM

- 7.2.1. Insertar eventos REM
- 7.2.2. Obtener eventos REM
- 7.2.3. Actualizar eventos REM
- 7.2.4. Eliminar eventos REM

7.3. Gestionar grupos REM

- 7.3.1. Insertar grupos REM
- 7.3.2. Obtener grupos REM
- 7.3.3. Actualizar grupos REM
- 7.3.4. Eliminar grupos REM

7.4. Generar reportes REM

2.3.3 Características no funcionales de la aplicación web

Al concebir las historias de usuario no se detallan ciertos aspectos que se deben precisar en la elaboración de una aplicación. Estos aspectos influyen en gran medida en el funcionamiento del producto final. La metodología XP orienta que los detalles de implementación de las historias de usuario se tienen en cuenta en el mismo momento de la concepción, por estas razones se describen a continuación de forma general, algunas de las características no funcionales que debe tener el sistema.

Los requerimientos no funcionales de un sistema son propiedades, cualidades o restricciones que el producto debe cumplir, tales como restricciones de tiempo, estándares de desarrollo, entre otros con el objetivo de lograr un producto confiable, atractivo y seguro (37).

Apariencia o interfaz externa

El sistema deberá poseer una interfaz gráfica uniforme que incluirá un menú estructurado en pestañas, cada pestaña representa uno de los módulos que integran la aplicación, las funcionalidades de cada módulo se encuentran en un panel ubicado a la izquierda en cada vista del sistema.

Requerimientos de hardware

Teniendo como base los requisitos mínimos de hardware de las herramientas y tecnologías seleccionadas se plantea:

Para el uso del cliente: PC Pentium IV o superior, CPU Intel Celeron a 1.8 GHZ o superior, 512 MB de RAM o superior y 650 MB de espacio libre en disco.

Para el uso en el servidor: PC con CPU Intel Celeron a 1.8 GHZ o superior, 512 MB de RAM o superior y 40 GB de espacio libre en disco.

Requerimientos de seguridad

La seguridad del sistema desarrollado está basada en niveles de acceso sobre las funcionalidades y la información manejada por el sistema. Los principios que determinan la seguridad en el sistema son los siguientes:

- La seguridad se establecerá por roles que serán asignados a los usuarios que interactúan con el sistema, garantizando de esta forma que la información almacenada solo sea modificada o visualizada por los usuarios autorizados.

Requerimientos de Usabilidad

- El sistema podrá ser utilizado por personas que tengan conocimientos básicos en el manejo de las computadoras, en este caso será utilizado por la Vicedecana de Extensión y Residencia de la Facultad 1, profesores, estudiantes de la Facultad 1, invitado e instructores de la Residencia 2.
- El sistema utilizará el idioma español para los mensajes y textos de la interfaz.
- La navegabilidad no debe ser muy compleja, todas las funcionalidades deben ser rápidamente accesibles por el usuario.

- El sistema debe ser escalable para que al agregar nuevas funcionalidades no sean afectadas las que ya están funcionando.

Requerimientos de Disponibilidad.

El sistema debe estar disponible las 24 horas del día exceptuando solo los días licenciados para mantenimiento.

Portabilidad.

La aplicación es multiplataforma (Windows/Linux).

2.3.4 Historias de usuario

Las HU sustituyen a los documentos de especificación funcional, y a los “casos de uso”. Estas “historias” son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. La diferencia más importante entre estas historias y los tradicionales documentos de especificación funcional se encuentra en el nivel de detalle requerido (38).

A continuación en la tabla 2 se muestra un ejemplo de la HU correspondiente a la funcionalidad “Gestionar evento artístico” del módulo festival de artistas aficionados. El resto de las HU se encuentran en los anexos del presente documento (Ver Anexo 6).

Tabla 2. HU_Gestionar evento artístico

Historia de usuario	
ID: 1	Nombre: gestionar evento artístico.
Usuario: administrador	
Prioridad en el negocio: alta	Riesgo en desarrollo: medio
Estimación aproximada: 7	Iteración asignada: 1
Descripción: permite insertar, obtener, eliminar y actualizar los eventos artísticos en el festival de artistas aficionados.	
Observaciones: responde a la HU Gestionar evento artístico.	

El propósito general de la fase de planificación es que los clientes y desarrolladores lleguen a un acuerdo sobre cuales historias de usuario deben estar listas para la primera liberación. De forma particular, es en esta fase donde los programadores obtienen la estimación del esfuerzo necesario para la elaboración de las historias de usuario. La medida para calcular dicho esfuerzo es el punto y por lo general, un punto equivale a una semana ideal, en el caso particular de la presente investigación se mantendrá durante todo el documento como valor de un punto, un día ideal debido a las características del equipo de desarrollo y el entorno de trabajo.

El tiempo ideal de desarrollo es aquel en el cual se escribe el código sin distracciones, logrando dedicarse a tiempo completo a esta actividad. Dicha estimación incluye todo el esfuerzo asociado a la implementación de la historia de usuario, como son: las pruebas unitarias, integración, refactorización del código, la preparación y ejecución de las pruebas de aceptación, entre otras actividades.

Teniendo en cuenta los intereses del cliente se asigna la prioridad a cada historia de usuario, con vista a cumplir con uno de los principales objetivos que persigue la metodología de desarrollo XP, aumentar el valor del producto final en el menor tiempo de desarrollo posible. Es en esta fase se obtiene un aproximado del cronograma de entrega de cada una de las liberaciones (40).

2.3.5 Prioridad de las historias de usuario

La asignación de prioridad a las historias de usuario por el cliente es lo que decide el orden en el cual estas se implementarán, siempre y cuando estén de acuerdo usuario y desarrollador. Las historias de usuario se clasifican en baja, media y alta (40). Teniendo en cuenta la solicitudes del cliente y la lógica del negocio, en la tabla 3 se define la prioridad de cada historia de usuario:

Tabla 3. Prioridad de las historias de usuario.

Historia de usuario	Prioridad
Autenticar usuario	media
Gestionar personal	media
Gestionar evento artístico	alta
Gestionar obra	alta
Gestionar actividades colaterales FAA	alta

Generar reportes FAA	alta
Gestionar estructura deportiva	alta
Gestionar actividades colaterales JD	alta
Gestionar evento deportivo	alta
Gestionar agrupación deportiva	alta
Gestionar reportes JD	alta
Gestionar cuarterería	alta
Gestionar evaluación mensual en la residencia	alta
Generar reportes residencia	alta
Gestionar áreas TSU	alta
Gestionar eventos TSU	alta
Gestionar grupos TSU	alta
Generar reportes TSU	alta
Gestionar estructura de mando	alta
Gestionar eventos REM	alta
Gestionar grupos REM	alta
Generar reportes REM	alta

2.3.6 Estimación de esfuerzo de las historias de usuario

Una vez asignadas las prioridades de las historias de usuario, se prosigue a estimar el esfuerzo necesario para la elaboración de cada una de ellas por parte de los desarrolladores. Esta estimación se basa principalmente en la velocidad del equipo de desarrollo y en la semejanza con historias de usuario desarrolladas con anterioridad (41). Las HU de la presente investigación tienen un valor de 5 y 10 puntos (días ideales), los puntos estimados son expresados en días ideales, ha de tenerse en cuenta que muy pocas veces el cronograma se lleva a cabo exactamente como se planifica. El esfuerzo necesario para construir las historias de usuario está basado en la técnica de estimación para el desarrollo ágil (42), quedando conformada según se muestra en la tabla 4.

Tabla 4. Estimación del esfuerzo por historia de usuario.

Historia de usuario	Esfuerzo necesario (Puntos estimados)
Autenticar usuario	10
Gestionar personal	10
Gestionar evento artístico	7
Gestionar obra	7
Gestionar actividades colaterales FAA	7
Generar reportes FAA	7
Gestionar estructura deportiva	7
Gestionar actividades colaterales JD	5
Gestionar evento deportivo	5
Gestionar agrupación deportiva	5
Gestionar reportes JD	5
Gestionar cuartería	5
Gestionar evaluación mensual en la residencia	5
Generar reportes residencia	5
Gestionar áreas TSU	5
Gestionar eventos TSU	5
Gestionar grupos TSU	5
Generar reportes TSU	5
Gestionar estructura de mando	5
Gestionar eventos REM	5
Gestionar grupos REM	5
Generar reportes REM	5

Una vez descritas e identificadas las historias de usuario y estimado el esfuerzo necesario para el desarrollo de cada una de ellas, se procede a realizar la planificación de las etapas de implementación del sistema. Con este fin se define un plan que contiene las iteraciones a realizar y la relación de las historias

de usuario que serán implementadas y en qué orden para cada iteración, teniéndose en cuenta la duración de las mismas (43). En la primera iteración se crea la arquitectura de todo el sistema; esto es logrado seleccionando las historias que hacen cumplir la estructura para todo el sistema. El cliente selecciona las historias para cada iteración. Al final de la última iteración, el sistema estará listo para ser entregado y llevarlo a producción (44).

De los planteamientos anteriores se determina que para el desarrollo de la aplicación web en cuestión se necesita un total de tres iteraciones, teniendo en cuenta la prioridad asignada a cada HU en correspondencia con las solicitudes del cliente.

En la **primera iteración** se seleccionaron las HU de alta prioridad de acuerdo a la solicitud del cliente de manera que el tiempo de desarrollo de las iteraciones esté balanceado y se conforme un producto que aún sin terminar pueda mostrar funcionalidades importantes del sistema según se muestra en la tabla 5.

Tabla 5. Primera iteración

Iteración		
Numero: 1	H.U: Gestionar evento artístico, Gestionar obra, Gestionar actividades colaterales FAA, Generar reportes FAA, Gestionar estructura deportiva, Gestionar actividades colaterales JD, Gestionar evento deportivo, Gestionar agrupación deportiva, Gestionar reportes JD.	Duración total: 55
Descripción: se establece la arquitectura a utilizar en el desarrollo del sistema. Se desarrollan las historias de usuario de prioridad alta, conformando un producto que aún sin terminar puedan mostrar las funcionalidades del sistema, dará una primera vista al cliente que suministrará su criterio para la incorporación de nuevos elementos y modificación de los existentes.		

En la **segunda iteración** se integraron las restantes HU de alta prioridad para el negocio con el objetivo de ir incorporando las principales funcionalidades al sistema según se muestra en la tabla 6.

Tabla 6. Segunda iteración

Iteración		
Numero: 2	H.U: Gestionar cuartería, Gestionar evaluación mensual en la residencia, Generar reportes residencia, Gestionar áreas TSU, Gestionar eventos TSU, Gestionar grupos TSU, Generar reportes TSU.	Duración total: 35
Descripción: se desarrollan historias de usuario de prioridad alta. Se llevan a cabo los señalamientos hechos por el cliente sobre la primera iteración y al finalizar dicho proceso se contará con una aplicación que incorpora las principales funcionalidades determinadas según las exigencias del cliente.		

En la **tercera iteración** las HU de prioridad media. Finalizándose de esta manera todas las funcionalidades descritas para la aplicación lo que permite obtener un producto final en su versión 1.0. Se sumaron todos los puntos de esfuerzos correspondientes a cada HU, quedando conformadas según se muestra en la tabla 7.

Tabla 7. Tercera iteración

Iteración		
Numero: 3	H.U: Autenticar usuario, Gestionar personal, Gestionar estructura de mando, Gestionar eventos REM, Gestionar grupos TSU, Generar reportes REM.	Duración total: 40
Descripción: se desarrollan las dos historias de usuario de prioridad media y aquellas restantes de prioridad alta, se llevan a cabo los señalamientos hechos por el cliente sobre la primera y segunda iteración. Al finalizar dicho proceso se contará con las funcionalidades descritas en estas historias de usuarios, finalizándose de esta manera todas las funcionalidades descritas para la aplicación y obteniéndose el producto final, versión 1.0.		

2.3.7 Cronograma de liberación

En el momento de planificar la liberación de un software se debe llevar un balance de este, pues si se realiza muy pronto no se tendrán suficientes funcionalidades que ameriten dicha liberación, por otro lado esperar mucho tiempo, conlleva a que el software desarrollado quede atrás respecto a la competencia.

Existe una creencia sobre XP, donde se debe llevar a cabo una liberación cada vez que es terminada una iteración, las liberaciones son lanzadas con un producto funcionando en óptimas condiciones, pero esto no ocurre siempre y un ejemplo de un escenario donde no ocurre, es al momento de dividir historias de usuario lo cual conlleva a esperar varias iteraciones, de forma tal que al ser liberado, éste contenga todas las funcionalidades necesarias de la historia de usuario original. La planificación de la liberación es un esfuerzo unido entre el cliente y el desarrollador, el primero decide qué historias de usuario tienen la mayor prioridad y el segundo estima el tiempo que le llevará implementar las mismas.

Entre los elementos a tener en cuenta en el cronograma de liberación, es fundamental saber que no siempre un plan fluye respecto a lo acordado, por esa razón el plan de liberación está en constante cambio. Un ejemplo de ello se tiene, cada vez que el usuario cambia la prioridad de una historia, cuando el desarrollador divide o une historias de usuario, aparecen imprevistos en las tecnologías a utilizar, entre

otras acciones. Es por esto que los cambios deben ser aceptados (45). El cronograma de liberación de la presente investigación queda conformado según se muestra en la tabla 8.

Tabla 8. Cronograma de liberación.

Iteración	Fecha de liberación
Primera iteración	4ta semana Febrero de 2015
Segunda iteración	4ta semana Marzo de 2015
Tercera iteración	1era semana Mayo de 2015

2.4 Patrones

Los patrones constituyen soluciones a problemas frecuentes que ocurren en un marco de trabajo. Luego de llegar a la solución se encapsulan todas las variables y factores de la misma, lo cual conforma una guía para resolver una y otra vez el mismo problema. Entre sus características se encuentran: describir el problema de forma sencilla, describir el contexto en el que ocurre, puntualizar los pasos a seguir, hacer énfasis en los puntos fuertes y débiles de la solución y referir otros patrones asociados. En el presente epígrafe se hace referencia al patrón arquitectónico utilizado para conformar el diseño de la aplicación propuesta (46).

2.4.1 Patrón arquitectónico

Un patrón es una solución reutilizable en problemas con un contexto común en el diseño de software. Se puede conceptualizar como una receta, pasos concretos a seguir que ayudarán a resolver un problema, puede ser considerado un patrón de diseño, puesto que deben cumplir ciertos objetivos o tener ciertas características como (47):

- Reutilizable en distintas circunstancias de diseño.
- Efectivo en problemas similares.
- Permite evitar riesgos que impidan reutilizar código.
- Simplifican la forma de documentar una aplicación.

MVC (*Model-View-Controller* o Modelo-Vista-Controlador) es un patrón de diseño que separa los datos, la lógica de negocios y las interfaces de usuario. Como su nombre lo dice, está separado en tres

componentes: modelo, controlador y vista. Está basado en la ideología de separación de conceptos y cumple perfectamente con los objetivos de los patrones de diseño según se muestra en la figura 1.

Modelo

Es la capa encargada de los datos, es decir, tiene mecanismos para acceder a la información y también para actualizar su estado. Comúnmente se encarga de comunicarse con la base de datos mediante funciones que accederán a las tablas y realizarán las funciones habituales de datos.

Vista

Se trata del código que permitirá presentar los datos que el modelo proporciona, como ejemplo se podría decir que en una aplicación web es el código HTML que permite mostrar la salida de los datos procesados.

Controlador

Es la capa que sirve de enlace entre la vista y el modelo. Envía comandos al modelo para actualizar su estado, y a la vista correspondiente para cambiar su presentación, pero no es el encargado de manipular los datos ni de generar una salida.

Partiendo de la descripción anterior el patrón seleccionado para el diseño de la aplicación es el MVC.

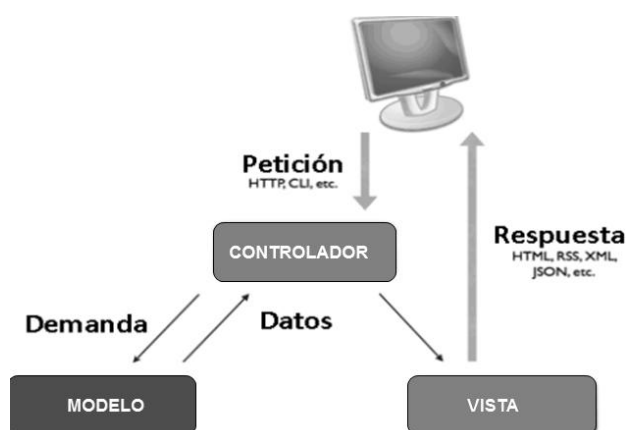


Figura 1: Modelo Vista Controlador

2.5 Diseño del sistema

Para el diseño de las aplicaciones, la metodología XP no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML, en su lugar la metodología XP utiliza ciertas técnicas como las tarjetas Clase, Responsabilidad y Colaboración (CRC).

2.5.1 Tarjetas CRC

XP estimula el uso de tarjetas CRC como un mecanismo eficaz para pensar en el software en un contexto orientado a objetos. Las tarjetas CRC identifican y organizan las clases orientadas a objetos que son relevantes para el incremento actual de software (48).

A continuación en la tabla 9 se muestra un ejemplo de las tarjetas CRC correspondiente a la HU “Gestionar evento artístico” con su clase evento_faa, el resto de las tarjetas CRC se encuentran en los anexos del presente documento (Ver Anexo 7).

Tabla 9. CRC Gestionar FAA clase: evento_faa

Tarjeta CRC	
Clase: evento_faa	
Responsabilidades	Colaboraciones
Listar evento artístico Crear evento artístico Modificar evento artístico Eliminar evento artístico	Evento_faa_lib.php

2.6 Diseño de la base de datos

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y utilizados por los sistemas de información de una empresa o negocio en particular.

Características

Entre las principales características de los sistemas de base de datos se encuentran:

- Independencia lógica y física de los datos.

- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar (49).

A continuación la figura 2 muestra el diagrama entidad-relación de la base de datos que se utilizó, dicho diagrama se presenta fraccionado por módulos debido al tamaño de la base de datos manejada, en este caso se muestra el módulo de “Gestionar festival de artistas aficionados”, el resto de los diagramas se encuentran en los anexos del presente documento (Ver Anexo 8).

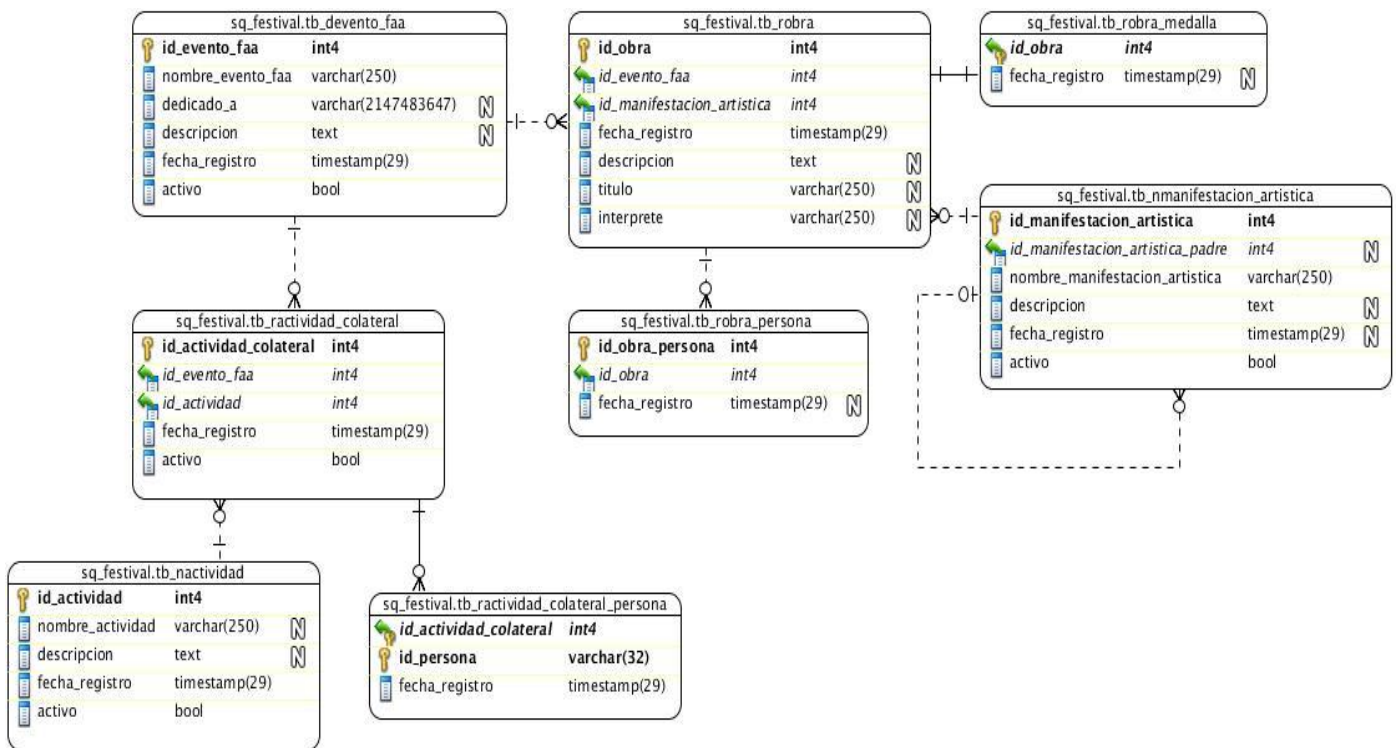


Figura 2: Módulo Gestionar FAA.

Capítulo 3. Implementación y pruebas del sistema

3.1 Introducción

El presente capítulo se abordan temas como el diseño de la aplicación, las tareas de ingeniería, diagrama de despliegue, así como la realización de las pruebas de software que permitan un exitoso desarrollo de la aplicación, las tareas de esta fase contribuyen a definir la arquitectura base de la aplicación así como el diseño de sus componentes.

3.2 Patrones de diseño

Los patrones de diseño son descripciones de clases y objetos relacionados que están particularizados para resolver un problema de diseño general en un determinado contexto (50). Entre los más conocidos se encuentran los patrones de Asignación de Responsabilidades y los patrones de la Banda de los Cuatro.

Patrones GRASP

Los Patrones Generales de Asignación de Responsabilidades de Software (GRASP) describen los principios fundamentales de la asignación de responsabilidades a objetos, de forma tal que se pueda diseñar software orientado a objetos. Estos no introducen ideas novedosas, sino que son la mera codificación de los principios básicos más usados. Los patrones GRASP están compuestos por: Experto, Creador, Bajo Acoplamiento, Alta Cohesión, Controlador, entre otros (51).

Patrones GOF

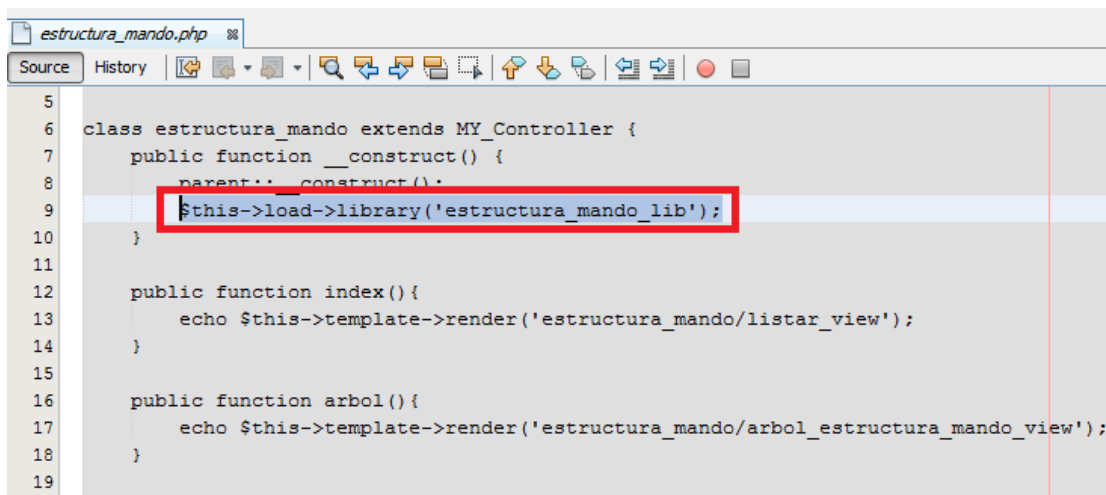
Los patrones de la Banda de los Cuatro (GoF por sus siglas en inglés), se encuentran agrupados en 3 divisiones en dependencia de su responsabilidad, estas son: comportamiento, creacionales y estructurales (52). En el diseño de la presente investigación para el desarrollo de subsistemas más pequeños se utiliza el *Singleton*.

En el desarrollo de la presente aplicación se evidencia el empleo de los siguientes patrones de diseño GOF:

Singleton: este patrón lo implementa CodeIgniter por defecto pues todos sus controladores constituyen un Singleton.

El patrón de diseño **Singleton** (instancia única) está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto (53).

En la clase controladora `estructura_mando` en su constructor instancia la librería `estructura_mando_lib.php`, de manera que no hay necesidad de instanciar esta librería en cualquier otra parte del programa. Su intención consiste en garantizar que una clase solo tenga una instancia y proporcionar un punto de acceso global a ella. Para asegurar que la clase no pueda ser instanciada nuevamente se regula el alcance del constructor. A continuación en la figura 3 se muestra un ejemplo de este patrón.



```
5
6 class estructura_mando extends MY_Controller {
7     public function __construct() {
8         parent::__construct();
9         $this->load->library('estructura_mando_lib');
10    }
11
12    public function index(){
13        echo $this->template->render('estructura_mando/listar_view');
14    }
15
16    public function arbol(){
17        echo $this->template->render('estructura_mando/arbol_estructura_mando_view');
18    }
19
```

Figura 3: Patrón *Singleton*.

En el desarrollo de la presente aplicación se evidencia el empleo de los siguientes patrones de diseño GRASP:

Controlador: este patrón se pone de manifiesto en los controladores de la aplicación que son los encargados de recibir la petición, ejecutar esta petición y enviar la respuesta correspondiente a la vista. En la figura 4, el menú de la izquierda define cuales son las clases controladoras del marco de trabajo que recibirán las peticiones realizadas a la aplicación.

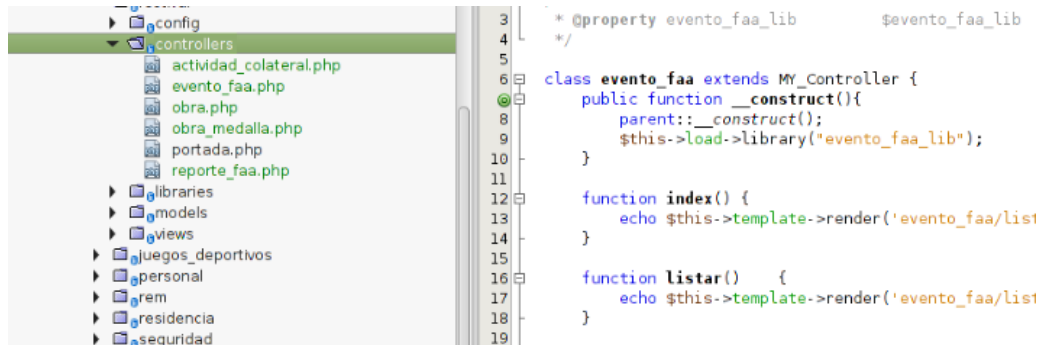


Figura 4: Patrón Controlador.

Alta cohesión: estos patrones vienen implementados en el propio framework CodeIgniter. Las librerías reflejan la alta cohesión pues la información que almacena cada una de estas clases es coherente y está relacionada con ella. La librería `jd_lib.php` es ella y solo ella la única encargada de la lógica de esa parte del negocio. La figura 5 muestra un ejemplo del patrón alta cohesión.

```
<?php
class evento_jd_lib {
    public function __construct(){
        $this->_ci =& get_instance();
        $this->_ci->load->model('tb_devento_jd_md1');
    }
    public function obtenerEventosJd($inicio = null, $limite = null, $elementoOrdenar
        return $this->_ci->tb_devento_jd_md1->obtenerEventosJdPorPagina($inicio, $limi
    }
    public function obtenerCantidadEventosJd($filtros = array()) {
        return count($this->obtenerEventosJd(null, null, null, null, $filtros));
    }

    public function obtenerEventoJdDadoId($id_evento_jd) {
        return $this->_ci->tb_devento_jd_md1->obtenerEventoJdDadoId($id_evento_jd);
    }
}
```

Figura 5: Patrón Alta cohesión.

Bajo acoplamiento: se evidencia en el hecho de que una clase no depende de muchas otras, lo cual potencia la reutilización y disminuye la dependencia entre estas. Las librerías reflejan el bajo acoplamiento. Un ejemplo es la librería `jd_lib` en su implementación solo contiene la lógica de su negocio y el acceso a datos con los modelos, esta clase no tiene instancia de más ninguna clase que refleje dependencias. A continuación la figura 6 muestra un ejemplo de la utilización de este patrón.

```
<?php
class evento_jd_lib {
    public function __construct() {
        $this->_ci =& get_instance();
        $this->_ci->load->model('tb_devento_jd_mdl');
    }
    public function obtenerEventosJd($inicio = null, $limite = null, $elementoOrdenar)
        return $this->_ci->tb_devento_jd_mdl->obtenerEventosJdPorPagina($inicio, $limi
    }
    public function obtenerCantidadEventosJd($filtros = array()) {
        return count($this->obtenerEventosJd(null, null, null, null, $filtros));
    }

    public function obtenerEventoJdDadoId($id_evento_jd) {
        return $this->_ci->tb_devento_jd_mdl->obtenerEventoJdDadoId($id_evento_jd);
    }
}
```

Figura 6: Patrón Bajo acoplamiento.

Creador: este patrón se pone de manifiesto en la clase `evento_faa_lib`, pues esta es la responsable de la creación de nuevos objetos o clases. Es aquí donde recae la responsabilidad de manipular o crear una vista o modelo. A continuación la figura 7 muestra un ejemplo de este patrón.

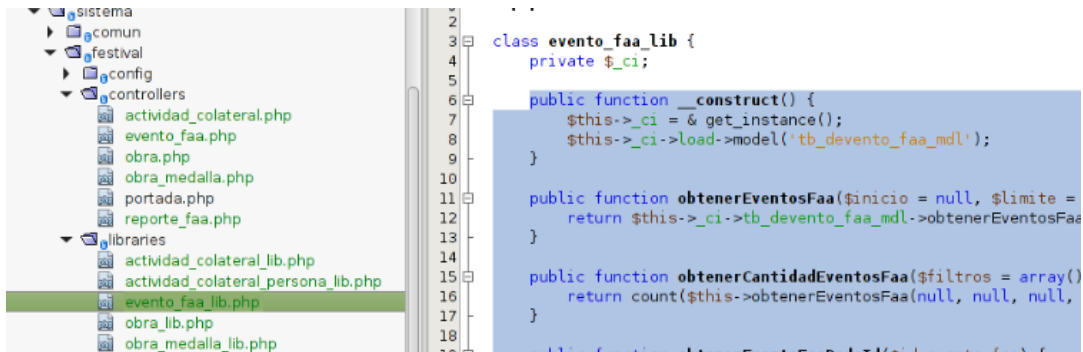


Figura 7: Patrón creador.

Mediador: se evidencia en las librerías utilizadas en la aplicación, pues estas actúan como una capa intermedia entre la capa de acceso a datos y el controlador, implementando la lógica del negocio. En la figura 8 se muestra un ejemplo del uso de este patrón.



Figura 8: Patrón mediador

3.3 Implementación del sistema

Las tareas llevadas a cabo en la fase de implementación, tienen como objetivo detallar las HU, lo cual facilita el entendimiento en el proceso de implementación. Cada HU puede contener una o más tareas de ingeniería, explicando de forma general las acciones que se realizan en la misma.

3.3.1 Tareas de las historias de usuario

A continuación en la tabla 10 se muestra un ejemplo de las tareas de ingenierías llevadas a cabo en el desarrollo de la aplicación, el resto de las tareas se encuentran en los anexos del presente documento (Ver Anexo 9).

Tabla 10. Tarea: Diseñar la vista de gestionar evento artístico.

Tareas de Historia de Usuario		
ID: 1	H.U: Gestionar evento artístico.	Iteración : 1
Nombre: diseñar la vista de gestionar evento artístico.		
Tipo: desarrollo	Puntos estimados: 2	
Fecha inicio: 7/1/2015	Fecha fin: 9/1/2015	
Responsable: Lidia del Carmen Leyva Feijoó y Alejandro López Rodríguez.		
Descripción: desarrollar la interfaz de usuario de la primera vista del evento artístico del festival de artistas aficionados, centrándose en el formulario de gestión de evento artístico.		

3.4 Estándares de codificación

Los estándares de codificación establecen las pautas de programación enfocadas en la estructura y apariencia física del código para facilitar la legibilidad, comprensión y mantenimiento del mismo. Dichos estándares definen la nomenclatura de las variables, objetos, métodos, funciones y su principal objetivo es llevar a cabo la realización de buenas prácticas y recomendaciones de diseño, para lograr mayores niveles de calidad en los productos de software (54).

A continuación se describe la nomenclatura para las clases, objetos, funciones y variables utilizadas durante la implementación del sistema.

3.4.1 Nomenclatura de las clases

Los nombres de las clases comienzan con la primera letra en minúsculas, si el nombre es compuesto se separan las palabras con el carácter underscore “_” y la primera letra de la palabra subsiguiente se escribe en minúsculas igualmente como se muestra en la figura 9.

```
class actividad_colateral extends MY_Controller {  
  
    public function __construct() {  
        parent::__construct();  
        $this->load->library("actividad_colateral_lib");  
        $this->load->library("evento_faa_lib");  
        $this->load->library("actividad_colateral_persona_lib");  
    }  
}
```

Figura 9: Nomenclatura de las clases.

3.4.2 Nomenclatura de las funcionalidades

Para la nomenclatura de las funcionalidades o métodos se utiliza el estándar Camel Case, el mismo consiste en que si el nombre de la funcionalidad tiene una sola palabra, esta se escribe con la primera letra en minúsculas, y si es compuesta, la primera letra de cada una de las palabras subsecuentes se escribe en mayúsculas (55). En la figura 10 se muestra un ejemplo.

```
public function registrarColateralJd($datos) {  
    $datos_colateral_jd = array();  
    $datos_colateral_jd['id_evento_jd'] = $datos['id_evento_jd'];  
    $datos_colateral_jd['id_actividad_jd'] = $datos['id_actividad_jd'];  
  
    return $this->_ci->tb_ractividad_colateral_jd_mdl->registrar($datos_colateral_jd, TRUE);  
}
```

Figura 10: Nomenclatura de las funcionalidades.

3.4.3 Nomenclatura de las variables

Los nombres de las variables y los objetos se escriben con la primera letra en minúsculas, y si el nombre es compuesto, se separa cada palabra utilizando el “_”, como se muestra en la figura 11.

```
public function registrarColateralJd(){  
    $verificar = array('id_evento_jd', 'id_actividad_jd', 'estudiantes_asociados');  
    if ($this->input->is_post_back($verificar)) {  
        $all_post = $this->input->all_post();  
        $this->db->trans_start();  
    }  
}
```

Figura 11: Nomenclatura de las variables.

3.5 Pruebas

Dentro de las buenas prácticas de la metodología XP, se encuentra la de llevar a cabo un Desarrollo Guiado por Pruebas, pues de esta manera se reduce el número de errores no detectados, así como el tiempo entre la introducción de estos en el sistema y su descubrimiento. Es por esta razón que el sistema en desarrollo está siendo testeado constantemente. La metodología XP divide las pruebas en 2 grupos fundamentales, las pruebas unitarias y las pruebas de aceptación (56).

3.5.1 Pruebas unitarias (PU)

Los pruebas unitarias o test unitarios son los más importantes para el practicante. Cada test unitario es un paso de avance en el camino de la implementación del software. Todo test unitario debe ser: atómico, independiente y rápido. Si no cumple estas premisas entonces no es un test unitario, aunque se ejecute con una herramienta tipo *xUnit*. Para conseguir cumplir estos requisitos, un test unitario aísla la parte del programa que necesita ejecutar, de tal manera que el resto está inactivo durante la ejecución (57).

Capítulo 3. Implementación y pruebas del sistema

A continuación en la tabla 11 se muestra un ejemplo de las pruebas estructurales de caja blanca aplicadas al sistema, en este caso para la funcionalidad “existe evento FAA”, el resto de las pruebas se encuentran en el anexo del presente documento (ver anexo 11).

Tabla 11. Prueba estructural de caja blanca para la funcionalidad: existe evento FAA.

Prueba estructural de caja blanca.	Código de caso de prueba.												
	Existe evento FAA.												
Probador:	Alejandro López Rodríguez.												
Tipo de dato esperado:	Array.												
Código al que se aplica:	<pre> 28 public function existeEventoFaa(\$datos, \$modificar = false) { 29 \$bandera = false; 30 \$arr = array(); 31 \$arr['nombre_evento_faa'] = \$datos['nombre_evento_faa']; 32 33 if (\$modificar == true) { 34 35 \$evento = \$this->_ci->tb_deevento_faa_mdml->obtenerDadoAtributos(\$arr); 36 37 if (!empty(\$evento)) { 38 foreach (\$evento as \$objP) { 39 if (\$objP->id_evento_faa != \$datos['id_evento_faa']) { 40 \$bandera = true; 41 } 42 } 43 } 44 else 45 \$bandera = false; 46 } else { 47 48 \$evento = \$this->_ci->tb_deevento_faa_mdml->obtenerDadoAtributos(\$arr); 49 \$contador = count(\$evento); 50 51 if (\$contador == 0) 52 \$bandera = false; 53 else 54 return true; 55 } 56 return \$bandera; 57 } </pre>												
Caso de prueba para los caminos básicos													
Tipo de dato esperado:	Array												
Función de evaluación:													
<table border="1"> <tr> <td>Test Name</td> <td>Existe evento faa</td> </tr> <tr> <td>Test Datatype</td> <td>Boolean</td> </tr> <tr> <td>Expected Datatype</td> <td>Boolean</td> </tr> <tr> <td>Result</td> <td>Passed</td> </tr> <tr> <td>File Name</td> <td>/var/www/lidia/base/application/sistema/festival/controllers/evento_faa.php</td> </tr> <tr> <td>Line Number</td> <td>16</td> </tr> </table>		Test Name	Existe evento faa	Test Datatype	Boolean	Expected Datatype	Boolean	Result	Passed	File Name	/var/www/lidia/base/application/sistema/festival/controllers/evento_faa.php	Line Number	16
Test Name	Existe evento faa												
Test Datatype	Boolean												
Expected Datatype	Boolean												
Result	Passed												
File Name	/var/www/lidia/base/application/sistema/festival/controllers/evento_faa.php												
Line Number	16												
Evaluación del caso de prueba:	Satisfactoria												

3.5.2 Pruebas de aceptación (PA)

Las pruebas de aceptación o test de aceptación permiten comprobar que el software cumple con los requisitos del negocio, una funcionalidad escrita con el lenguaje del cliente pero que puede ser ejecutada por la máquina. Estas pruebas son creadas a partir de las historias de usuario. Estas son el punto de partida del desarrollo en cada iteración (57). Las pruebas de aceptación permiten además, comprobar que la funcionalidad que se está probando sea la esperada por el cliente. Este tipo de pruebas funcionan como una caja negra pues cada una de ellas representa una salida esperada del sistema, donde es responsabilidad del cliente verificar la corrección de las pruebas y tomar decisiones.

A continuación en la tabla 12 se muestra la prueba de aceptación “Gestionar evento artístico”, el resto de las pruebas se encuentran en los anexos del presente documento (Ver Anexo 10).

Tabla 12. PA_Gestionar evento artístico

Prueba de aceptación		
Código: H.U3_P.A3	H.U: Gestionar evento artístico	
Nombre: Gestionar evento artístico		
Descripción: se maneja la información relacionada con los eventos artísticos del FAA.		
Condiciones de Ejecución: el usuario debe estar autenticado previamente y en dependencia del rol podrá manejar la información relacionada con los eventos artísticos.		
Entrada / Pasos de ejecución: se muestra un listado de los eventos artísticos que existen en la aplicación, además de permitir insertar, obtener, actualizar y eliminar los eventos artísticos en dependencia del rol asignado.		
Resultados esperados:	Escenario positivo: datos correctos	Escenario negativo: datos incorrectos
	Si los datos introducidos son correctos, el usuario puede terminar el proceso satisfactoriamente y se guarda la información en la base de datos, además se muestra un listado con los eventos artísticos ya creados.	Si los datos introducidos no son correctos, el usuario no puede terminar el proceso y el sistema muestra un mensaje de error.
Evaluación: satisfactoria		

3.5.3 Prueba funcional (PF)

Objetivo:

Se asegura el trabajo apropiado de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados.

Las metas de estas pruebas son:

1. Verificar el procesamiento, recuperación e implementación adecuada de las reglas del negocio.
2. Verificar la apropiada aceptación de datos.

Enfoque:

Los requisitos funcionales y las reglas del negocio.

Técnica:

Caja Negra. Se ejecuta cada función, usando datos válidos e inválidos, para verificar lo siguiente:

1. Que se aplique apropiadamente cada regla de negocio.
2. Que los resultados esperados ocurran cuando se usen datos válidos.
3. Que sean desplegados los mensajes apropiados de error y precaución cuando se usan datos inválidos (58).

A continuación en la tabla 13 se muestra un ejemplo de prueba funcional en este caso las pruebas de acceso al sistema.

Tabla 13. PF_Acceso al sistema.

Prueba	Entrada o Acción de usuario	Resultado esperado del Sistema	Confirmación
P.1	usuario y contraseña correcta.	El sistema permite el acceso al usuario, identificándolo correctamente y mostrándole la pantalla de bienvenida.	Si
P.2	usuario incorrecto y contraseña correcta.	El sistema niega el acceso y muestra la página de entrada nuevamente, mostrando	Si

		el mensaje “usuario y contraseña incorrectos”	
P.3	usuario correcto y contraseña incorrecta.	El sistema niega el acceso y muestra la página de entrada nuevamente, mostrando el mensaje “usuario y contraseña incorrectos”	Si
P.4	usuario y contraseña incorrectos.	El sistema niega el acceso y muestra la página de entrada nuevamente, mostrando el mensaje “usuario y contraseña incorrectos”	Si

3.5.4 Pruebas de seguridad (PS)

Objetivo

Nivel de Seguridad de la aplicación:

- Verifica que un actor solo pueda acceder a las funcionalidades y datos que le sean permitidos.

Nivel de Seguridad del sistema:

- Verificar que solo puedan acceder a la aplicación aquellos actores que tengan permiso.

Áreas

Seguridad del sistema, incluyendo acceso a datos o funcionalidades del negocio. Seguridad del sistema, incluyendo ingresos y accesos remotos al sistema.

Garantiza

Que los usuarios estén restringidos a funciones específicas o su acceso esté limitado únicamente a los datos que están autorizados a acceder. Que solo aquellos usuarios autorizados a acceder al sistema son capaces de ejecutar las funciones del sistema.

Técnicas

Identificar cada tipo de usuario, las funciones y datos a los que se debe autorizar.

En la tabla 14 se muestran ejemplos de los resultados obtenidos de diferentes pruebas de seguridad aplicadas al sistema.

Tabla 14. PS_Prueba de seguridad.

Prueba	Ejemplos de Pruebas	Resultado esperado del Sistema	Confirmación
P.1	Después de autenticarse en el sistema si el usuario tiene el rol de Administrador.	Tiene acceso a interactuar con todas las funcionalidades del sistema.	Si
P.2	Después de autenticarse en el sistema si el usuario tiene el rol de Estudiante.	Permite al estudiante tener un resumen de su recorrido en el área de la residencia.	Si
P.3	Se le realizaron ataques de inyección SQL al sistema, por ejemplo al formulario "crear evento deportivo", intencionalmente se introdujo la consulta SQL (delete from sq_juegos_deportivos.tb_devento_deportivo) en el campo "nombre".	El sistema no ejecuta la consulta	Si

3.5.5 Pruebas de carga y estrés

El rendimiento de la aplicación es un elemento clave en la aceptación de la misma, cualquiera que esta sea. Este es totalmente dependiente de varios factores, entre los que se incluyen, la calidad de la conexión a la red así como el hardware que se utilice.

El Apache JMeter fue la herramienta utilizada para realizar las pruebas de rendimiento a la aplicación web. Esta aplicación, entre su gama de funcionalidades recoge un número importante de datos sobre los resultados capturados. En la tabla 15 se presentan los datos arrojados bajo los siguientes valores:

Usuarios: 85.

Microprocesador: Un microprocesador Intel Celeron a 1.8 GHZ.

RAM: 512 MB.

De manera general se puede observar que las peticiones son ejecutadas en tiempos relativamente rápidos. Para todas las muestras de usuarios la ocurrencia de error se mantiene en 0.0 % lo que quiere decir que todas las peticiones realizadas se ejecutan satisfactoriamente.

Tabla 15. Datos asociados a prueba de carga y estrés.

	#muestras	Media	Mín	Max	Error	Rendimiento	kb/seg
Total	100	267	11	1222	0,00	66,6 kb/seg	217,0
	175	221	26	1566	0,00	135,7 kb/seg	473,0
	200	454	35	2556	0,00	168,3 kb/seg	537,0

- **Muestras:** cantidad de peticiones atendidas por el servidor.
- **Mínimo (Mín.):** indica el mínimo de tiempo de ejecución invertido para una petición con n usuarios haciendo peticiones de manera concurrente.
- **Máximo (Máx.):** indica el máximo de tiempo de ejecución invertido para una petición con n usuarios haciendo peticiones de manera concurrente.
- **Media:** representa el tiempo de ejecución promedio de una petición con n usuarios.
- **%Error:** indica la relación entre el total de peticiones y el número de peticiones que originaron errores.
- **Rendimiento (Pet./seg):** hace referencia al número de peticiones que el servidor puede procesar en un segundo.
- **Rendimiento (Kb./seg):** hace referencia a la cantidad de datos que el servidor puede procesar en un segundo.

3.5.6 Registro de no conformidades

Uno de los detalles que se deben de tener en cuenta una vez realizadas las pruebas son las no conformidades, las cuales se traducen en los errores encontrados y funcionalidades no deseadas por el cliente. Al final de cada iteración se le muestra al cliente una versión funcional del software de forma que pueda detectar aquellas no conformidades que serán corregidas al inicio de la subsiguiente iteración.

La figura 12 representa un gráfico de barras con los resultados de las pruebas desarrolladas en las tres iteraciones al sistema, asociadas a las no conformidades detectadas en cada una de ellas.

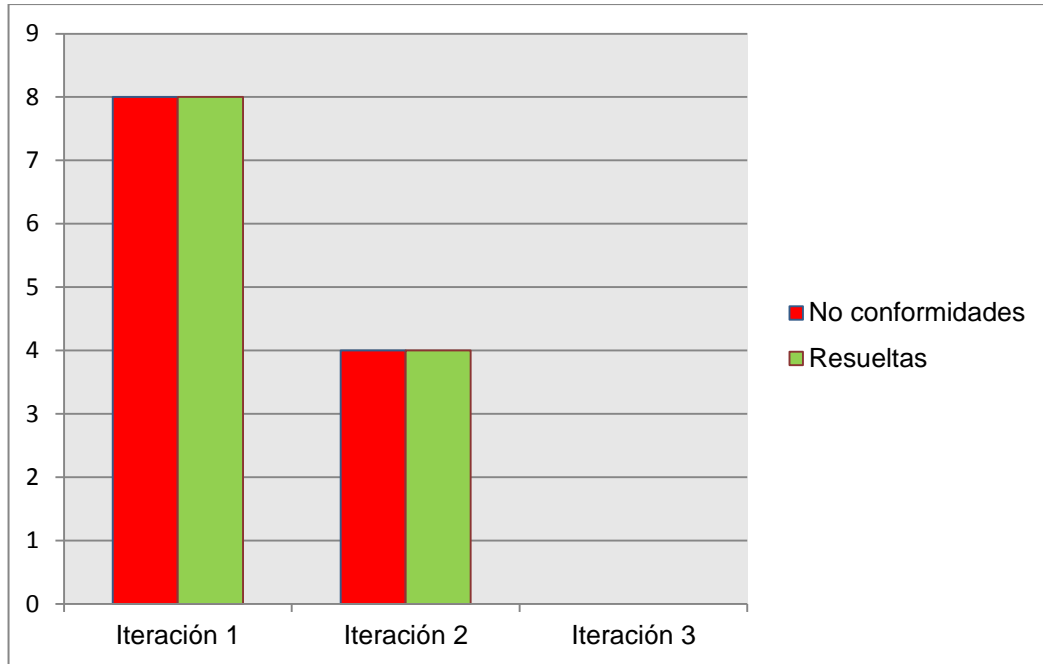


Figura 12: Resultados de las pruebas

Como parte de la metodología XP, las no conformidades encontradas en cada iteración son las primeras tareas a resolver de la iteración siguiente, siendo el cliente el encargado de ordenarlas por prioridad. Algunas de ellas al no ser críticas, son arrastradas a la siguiente iteración. Llevando a cabo este proceso, se logran minimizar los niveles de aceptación de errores. De esta manera quedaron resueltas las no conformidades detectadas en la aplicación desarrollada.

La tabla 16 muestra las no conformidades encontradas en cada una de las iteraciones.

Tabla 16. Registro de no conformidades.

No conformidades	
Iteración:	Descripción:
Primera	<ol style="list-style-type: none"> 1. Errores de falta de ortografía en los mensajes de la pantalla de autenticación. 2. Error al generar reportes asociados a la información de los estudiantes que han participado en los juegos deportivos. 3. Error al generar reportes asociados a la información de los estudiantes que han participado en el TSU. 4. El sistema no permite establecer una evaluación del mes de los estudiantes en el módulo de la residencia. 5. Errores de diseño.

	<ol style="list-style-type: none"> 6. Permite insertar números en el campo de nombre de usuario. 7. Retardo al cargar las pestañas. 8. No se generan los reportes asociados al módulo del TSU.
Segunda	<ol style="list-style-type: none"> 1. Pérdida de datos. 2. Errores de concordancia en los mensajes. 3. Mensajes de error poco claros. 4. Permite introducir número en campos donde solo se admiten letras.
Tercera	No se encuentran no conformidades en el sistema.

3.6 Diagrama de despliegue

Es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Correspondencia entre la arquitectura del software y la arquitectura del sistema. Cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo hardware similar. Los nodos poseen relaciones que representan medios de comunicación entre ellos. El modelo de despliegue puede describir diferentes configuraciones de red. La funcionalidad de un nodo se define por los componentes que se distribuyen en ese nodo (59).

El diagrama de la figura 13 muestra la distribución de los nodos necesarios para el despliegue de la aplicación web. Se ubicará una computadora cliente la cual se conecta con el servidor web a través del protocolo HTTPS, este establece una conexión con el servidor de base de datos mediante el protocolo TCP, permitiendo así una correcta comunicación entre los diferentes componentes.

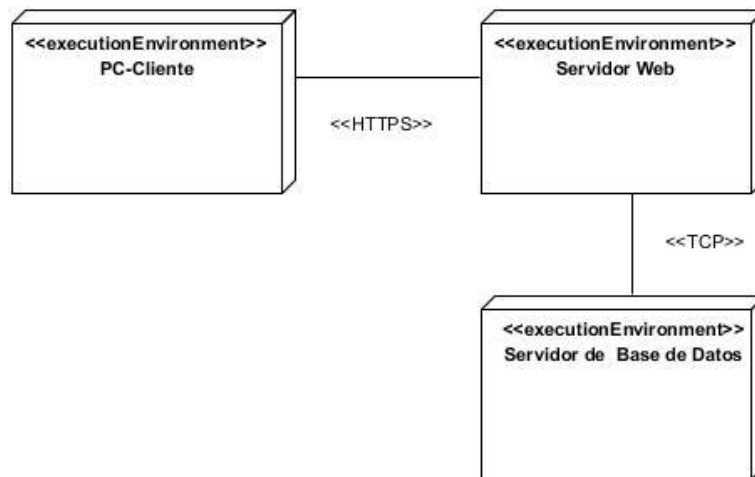


Figura 13: Diagrama de despliegue

Conclusiones generales

- El estudio realizado de los diferentes sistemas homólogos, demostró la necesidad de desarrollar en la Facultad 1 de la UCI una aplicación web que gestione los procesos extensionistas.
- Identificar y describir las HU, así como seleccionar la arquitectura y los patrones de diseño que se deben tener en cuenta para el desarrollo del sistema, permitió comprender las características que debía poseer la solución, facilitando en gran medida su implementación.
- La aplicación web desarrollada para el Vicedecanato de Extensión y Residencia de la Facultad 1, permitió gestionar de manera integrada los diferentes procesos extensionistas, posibilitando eliminar la pérdida o duplicación de la información. Los reportes generados por el sistema facilitan el análisis oportuno de la información para la toma de decisiones.

Recomendaciones

Como resultado del proceso de investigación se sugieren aspectos que serían recomendables a tener en cuenta para el futuro perfeccionamiento del sistema:

- Adaptar el sistema a entornos móviles.
- Extender el uso del sistema a todas las facultades de la Universidad de las Ciencias Informáticas.

Referencias bibliográficas

1. Informatización de la Sociedad. [En línea]. 22 enero 2013. [Consulta: 1 junio 2015]. Disponible en: http://www.ecured.cu/index.php/Informatizaci%C3%B3n_de_la_Sociedad
2. MARÍA ELENA DEL HUERTO MARIMÓN. La Extensión Universitaria como vía para fortalecer los vínculos Universidad-Sociedad desde la promoción de salud. [En línea]. 27 septiembre 2010. Disponible en: http://www.bvs.sld.cu/revistas/spu/vol33_02_07/spu05207.htm#cargo
3. La Modelación como método científico. [En línea]. 12 junio 2010. Disponible en: <http://es.scribd.com/doc/96367818/La-Modelacion-como-metodo-cientifico-2>
4. Gestión. *Definición MX* [En línea]. [Consulta: 24 octubre 2014]. Disponible en: <http://definicion.mx/gestion/>
5. Concepto de gestión. Qué es, Significado y Definición. [En línea]. [Consulta: 4 noviembre 2014]. Disponible en: <http://concepto.de/concepto-de-gestion/>
6. VENTAJAS DE TRABAJAR BAJO UN SISTEMA DE GESTIÓN. *Seguridad en Ambientes Laborales* [En línea]. 16 noviembre 2014. [Consulta: 16 noviembre 2014]. Disponible en: <http://seguridadyambiente.wordpress.com/2008/10/11/ventajas-de-trabajar-bajo-un-sistema-de-gestion/>
7. Proceso. *Definición MX* [En línea]. [Consulta: 4 noviembre 2014]. Disponible en: <http://definicion.mx/proceso/>
8. Tipos de procesos y sus características por Javier150. *Buenas Tareas* [En línea]. [Consulta: 4 noviembre 2014]. Disponible en: <http://www.buenastareas.com/ensayos/Tipos-De-Procesos-y-Sus-Caracteristicas/1612990.html>
9. PÉREZ LINDO, Augusto. Fundamentos de la gestión del conocimiento en la universidad. *www.gestuniv.com.ar/gu_08/v3n2a1.htm* [En línea]. [Consulta: 4 noviembre 2014]. Disponible en: http://www.gestuniv.com.ar/gu_13/v5n1a2.htm
10. CEDEÑO FERRÍN, Julio y FERRÍN, Julio Cedeño. Tendencias del proceso de gestión de la Extensión Universitaria y su impacto cultural. *Humanidades Médicas*. 18 diciembre 2012. Vol. 12, no. 3, p. 499-514.
11. Web. [En línea]. [Consulta: 5 noviembre 2014]. Disponible en: <http://www.mastermagazine.info/termino/7216.php>
12. Descripción de componentes: servidor web, SGBD y aplicaciones. — Institut Puig Castellar. [En línea]. [Consulta: 7 noviembre 2014]. Disponible en: <http://elpuig.xeill.net/Members/vcarceler/asix-m09/uf1/nf1/a1>

13. ZAMORA, Marlen. Aplicaciones Web: VENTAJAS Y DESVENTAJAS DE APLICACIONES WEB. *Aplicaciones Web* [En línea]. 6 octubre 2012. [Consulta: 7 noviembre 2014]. Disponible en: <http://unidad6aplicacionesweb1.blogspot.com/2012/10/blog-post.html>
14. S.L, AvaiBook On-Line. AvaiBook Sports - Plataforma de gestión de eventos deportivos. <https://www.avaibooksports.com/> [En línea]. 16 noviembre 2014. [Consulta: 16 noviembre 2014]. Disponible en: <https://www.avaibooksports.com/>
15. Christian Báez. Sistema de gestión patrimonial. [En línea]. 2006. Disponible en: <http://www.portalpatrimonio.cl/quienes-somos/>
16. LIBRO, Instituto Cubano del. Literatura Cubana - Portal Cubaliteraria. [En línea]. [Consulta: 16 noviembre 2014]. Disponible en: <http://www.cubaliteraria.com/>
17. POR, Publicado y HINOJOSA, rea López. Tipos de Servidores Web. [En línea]. [Consulta: 16 noviembre 2014]. Disponible en: <http://servidoreswebalh.blogspot.com/>
18. Apache, el servidor Web más reconocido. [En línea]. [Consulta: 16 noviembre 2014]. Disponible en: <http://www.empresayeconomia.es/aplicaciones-para-empresas/apache-el-servidor-web-mas-reconocido.html>
19. Lenguajes de lado servidor o cliente. *DesarrolloWeb.com* [En línea]. [Consulta: 16 noviembre 2014]. Disponible en: <http://www.desarrolloweb.com/articulos/239.php>
20. Qué es HTML. *DesarrolloWeb.com* [En línea]. [Consulta: 2 diciembre 2014]. Disponible en: <http://www.desarrolloweb.com/articulos/que-es-html.html>
21. HTML. *Mozilla Developer Network* [En línea]. [Consulta: 2 diciembre 2014]. Disponible en: <https://developer.mozilla.org/es/docs/Web/HTML>
22. 1.4. Cómo incluir JavaScript en documentos XHTML (Introducción a JavaScript). [En línea]. [Consulta: 2 diciembre 2014]. Disponible en: http://librosweb.es/javascript/capitulo_1/como_incluir_javascript_en_documentos_xhtml.html
23. ¿Que se puede decir del PHP? [En línea]. [Consulta: 2 diciembre 2014]. Disponible en: <http://www.programacionweb.net/articulos/articulo/que-se-puede-decir-php/>
24. Qué es PHP. *DesarrolloWeb.com* [En línea]. [Consulta: 2 diciembre 2014]. Disponible en: <http://www.desarrolloweb.com/articulos/392.php>
25. PostgreSQL - pgAdmin - Bases de Datos - Tutoriales. *Solvetic* [En línea]. [Consulta: 2 diciembre 2014]. Disponible en: <http://www.solvetic.com/tutoriales/article/299-postgresql-pgadmin/>
26. Manual del usuario de PostgreSQL. *Scribd* [En línea]. [Consulta: 4 diciembre 2014]. Disponible en: <https://es.scribd.com/doc/5703210/Manual-del-usuario-de-PostgreSQL>

27. XP - Extreme Programming Ingeniería de Software. [En línea]. [Consulta: 2 diciembre 2014]. Disponible en: http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programming.html
28. LETELIER, Patricio y LETELIER, Patricio. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). *www.cyta.com.ar/ta0502/v5n2a1.htm* [En línea]. 15 abril 2006. [Consulta: 2 diciembre 2014]. Disponible en: http://www.cyta.com.ar/ta0502/b_v5n2a1.htm
29. Modelo xp para desarrollo de proyecto. [En línea]. 00:45:46 UTC. [Consulta: 2 diciembre 2014]. Disponible en: <http://es.slideshare.net/johitaamiga/modelo-xp-para-desarrollo-de-proyecto>
30. Lenguaje de modelado de objetos. *Los diccionarios y las enciclopedias sobre el Académico* [En línea]. [Consulta: 2 diciembre 2014]. Disponible en: http://enciclopedia_universal.esacademic.com/31901/Lenguaje_de_modelado_de_objetos
31. MEZA, Mirna. Herramientas Case: I N T R O D U C C I Ó N. *Herramientas Case* [En línea]. sábado, de abril de 2011. [Consulta: 2 diciembre 2014]. Disponible en: <http://fds-herramientascase.blogspot.com/2011/04/i-n-t-r-o-d-u-c-c-i-o-n.html>
32. Visual Paradigm for UML (ME) - (Paradigma Visual para UML (ME)) (Visual Paradigm for UML (ME)) por Visual Paradigm International Ltd. - reporte y descarga. [En línea]. [Consulta: 2 diciembre 2014]. Disponible en: http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/
33. NetBeans IDE entorno de desarrollo para lenguajes como Java PHP C/C++ Groovy. [En línea]. [Consulta: 2 diciembre 2014]. Disponible en: <http://www.genbetadev.com/herramientas/netbeans-1>
34. CodeIgniter. *DesarrolloWeb.com* [En línea]. [Consulta: 2 diciembre 2014]. Disponible en: <http://www.desarrolloweb.com/articulos/codeigniter.html>
35. ¿Qué es jQuery? *Mexired* [En línea]. [Consulta: 2 diciembre 2014]. Disponible en: <http://www.mexired.com/blog/que-es-jquery/>
36. ING.DANAY PÉREZ RAMÍREZ. *METODOLOGÍAS ÁGILES.¿CÓMO DESARROLLO UTILIZANDO XP?* 2008.
37. Joe Luttrell. y MICHAEL DOBSON. LESSONS FROM HISTORY. Functional versus NonFunctional Requirements and Testing. [En línea]. febrero 2013. Disponible en: <http://www.lessons-from-history.com/node/83>.
38. ING.JOSÉ, Joskowicz. *Reglas y Prácticas en eXtreme Programming* [En línea]. 2 octubre 2008. Disponible en: <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
39. Metodologias para el desarrollo de aplicaciones web. [En línea]. 05:30:18 UTC. [Consulta: 7 noviembre 2014]. Disponible en: <http://es.slideshare.net/ARCANGEL2032/metodologias-para-el-desarrollo-de-aplicaciones-web-9419415>

40. PENADÉS, M^a Carmen. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). 2008.
41. MITRE-HERNÁNDEZ, Hugo A., ORTEGA-MARTÍNEZ, Edgar y LEMUS-OLALDE, Cuauhtémoc. Estimación y control de costos en métodos ágiles para desarrollo de software: un caso de estudio. *Ingeniería, investigación y tecnología*. septiembre 2014. Vol. 15, no. 3, p. 403-418.
42. NILDA M. PÉREZ OTERO. *Estimación para Proyectos Software* [En línea]. 2011. Disponible en: <https://sistinfii.files.wordpress.com/2011/03/siii11-07-estimacion3b3n-y-calendarizacion3b3n.pdf>
43. XP - Extreme Programming Ingeniería de Software. [En línea]. [Consulta: 13 marzo 2015]. Disponible en: http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programming.html
44. Fernández Escribano. *Introducción a Extreme Programming*. 12 septiembre 2002.
45. Jose Hilario. Metodologías ágiles en el desarrollo del software. febrero 2013.
46. GARCIA, Ic. francisco. *Patrones De Alexander a la Tecnología de Objetos* [En línea]. marzo 2010. Disponible en: <http://repositorio.grial.eu/bitstream/grial/356/1/patrones1.pdf>
47. MVC y su importancia en la Web. *michelletores.mx* [En línea]. 16 noviembre 2014. [Consulta: 16 noviembre 2014]. Disponible en: <http://michelletores.mx/mvc-y-su-importancia-en-la-web/>
48. Metodología XP. [En línea]. 2010. Disponible en: <https://sites.google.com/site/xpmetodologia/marco-teorico/funcionamiento>
49. Damián Pérez Valdés. Definición de Base de Datos. [En línea]. 2013. Disponible en: <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>
50. Intro a Patrones de Diseño(GoF). *Scribd* [En línea]. [Consulta: 7 abril 2015]. Disponible en: <https://es.scribd.com/doc/69490851/Intro-a-Patrones-de-Diseno-GoF>
51. ANDRÉS GROSSO. Patrones GRASP | Prácticas de Software. [En línea]. [Consulta: 7 abril 2015]. Disponible en: <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>
52. Patrones de diseño GoF. *GodTIC* [En línea]. [Consulta: 7 abril 2015]. Disponible en: <http://www.godtic.com/blog/2012/11/15/patrones-de-diseno-gof/>
53. Singletón. *Marco de Desarrollo de la Junta Andalucía*. 2012.
54. ING. VIRGEN DAMARIS QUEVEDO CAMPINS, MSC. MEYLIN MARTÍNEZ CHONG. CEDRUX. Solución para sistemas de control logístico. 2010.
55. Codificación estándar básica. [En línea]. Disponible en: <http://www.php-fig.org/psr/psr-1/es/>

56. MSC. TORRES. *PRUEBAS DEL SISTEMA EN PROGRAMACIÓN EXTREMA* [En línea]. Disponible en: http://www.lsi.us.es/~javierj/investigacion_ficheros/PSISEXTREMA.pdf
57. JURADO, CARLOS BLÉ. *Diseño ágil con TDD*. 2010.
58. ISABEL BLANK. *Pruebas de Funcionalidad* [En línea]. mayo 2015. Disponible en: http://carolina.terna.net/ingsw3/datos/Pruebas_Funcionales.pdf
59. 15. SALAMANCA, U. D. Y AUTOMÁTICA, D. D. I. Y. *Ingeniería del Software. Tema 6: Diseño orientado a objetos Última actualización* [En línea]. Disponible en: <http://es.scribd.com/doc/49708406/19/Modelo-de-despliegue>