



Universidad de las Ciencias
Informáticas

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

*Sistema Web de la Dirección de Extensión Cultural de la
Universidad de las Ciencias Informáticas*

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

Autor:

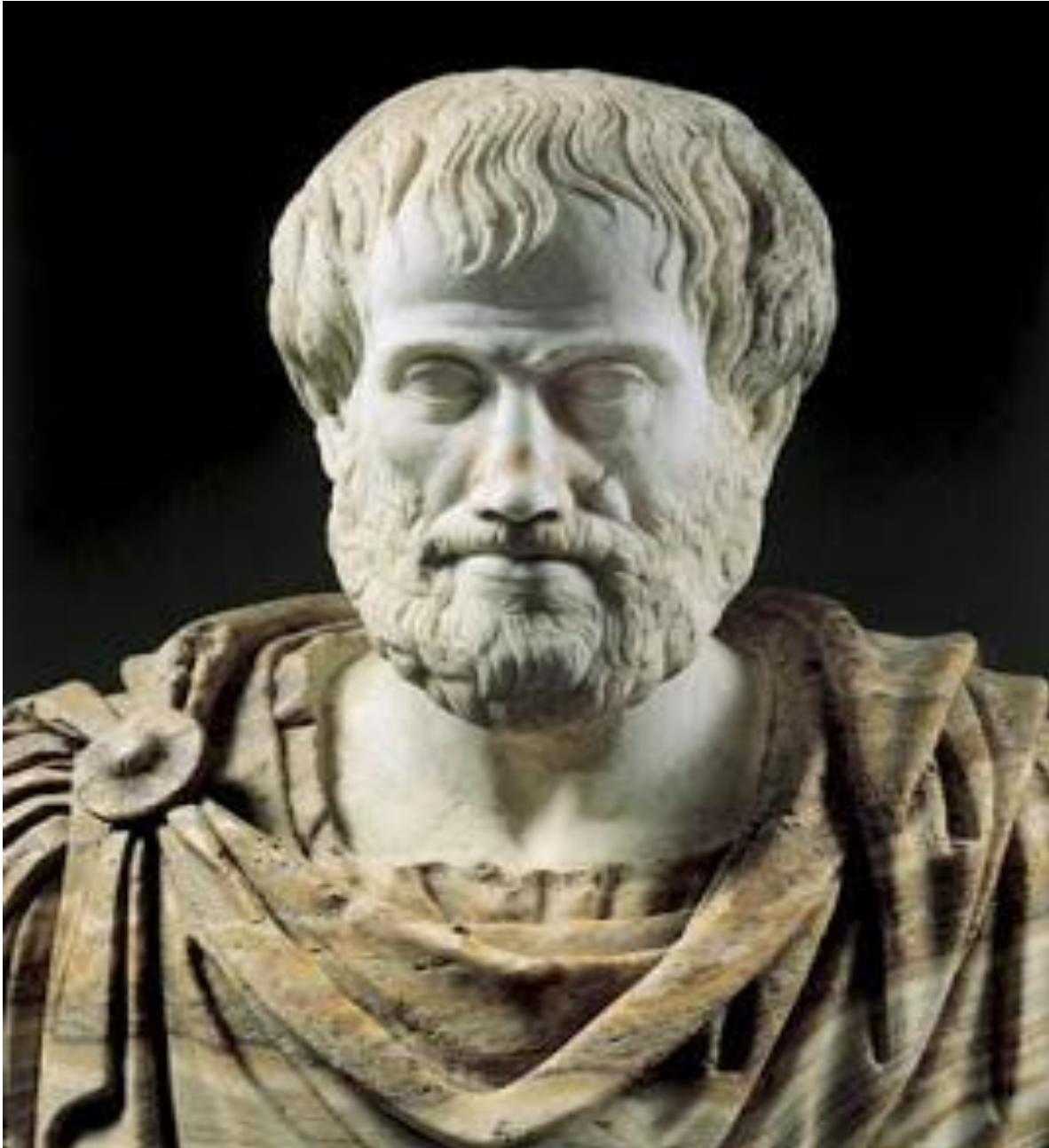
Adonis Vázquez Campos

Tutor:

MsC. Alién García Hernández

La Habana, junio de 2014

“Año 56 de la Revolución”



“Se quiere más lo que se ha conquistado con más fatiga”

Aristóteles

DECLARACIÓN DE AUTORÍA



Declaración de autoría

Declaro ser el autor del presente trabajo de diploma y reconozco a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo el presente a los __24__ días del mes de junio del año 2014.

Adonis Vázquez Campos
Firma del Autor

MsC. Alién García Hernández
Firma del Tutor

Datos de contacto

Autor

Adonis Vázquez Campos. Estudiante de Ingeniería en Ciencias Informáticas.

Correo electrónico: avcampos@estudiantes.uci.cu

Tutor

MsC. Alién García Hernández, Ingeniero en Ciencias Informáticas, Máster en Ciencias Matemáticas. Director de Extensión Cultural de la Universidad de las Ciencias Informáticas.

Correo electrónico: agarciah@uci.cu



Agradecimientos

Agradezco ante todo:

A mi tutor MsC. Alíen García Hernández por la comprensión y el apoyo brindado en el desarrollo de esta investigación.

A mi familia en general por ser la columna vertebral de mi educación, espíritu de lucha, por ser luz y guía en el camino de mi vida, por ser motivo de mi inspiración.

A la revolución cubana por ser el sustento de las nuevas generaciones.

A todos los que de una manera u otra, aportaron su granito de arena para llevar a cabo esta importante tarea.

Adonis Vázquez Campos.

Dedicatoria

Dedico la presente investigación:

A mis abuelos y padres: Luis Vázquez Pereña (Kíko) y Mirtha González Peña (Má), por su incondicional entrega, su comprensión infinita, su eterno amor e intachable firmeza, por ser guías imperecederos de mi destino y defensores incansables de la verdad, por haber sido, ser y continuar siendo personas admirables, intachables y dignos ejemplos a seguir.

A mi padre: Luis Vázquez González (Huesi), por ser ejemplo ante la vida, por su amor desmedido hacia la familia, por su inteligencia, por su solidaridad infinita, por su insuperable orgullo, por su tesón, abnegación y sentido de la responsabilidad, y porque me siento orgulloso de ser su hijo.

A mi madre: Carmen Rosa Campos, por traerme al mundo, por su dedicación en vida para con sus hijos, te quiero donde quiera que te encuentres.

A mi tío: Iglain Vázquez González (Igla), por su espíritu de lucha, su carácter firme, su incansable esfuerzo, su increíble carisma y sentido del humor, por su inteligencia práctica, por su amor hacia la familia, por ser ejemplo ante la toma de decisiones.

A mi tía: Tania Jiménez Calvo, por su responsabilidad, por su abnegación, por sus prontos, por quererme tanto y por ser la mejor de mis tías.

A mis hermanos: Darién (Chino), Rosalía (Rosy), Keyla, Melanie (Melón), Dalila (Lila), y primos: Mario Cesar Vázquez Jiménez (Mayito) y Adisnuvia Vázquez Jiménez (Macuca), para que sigan mi ejemplo.

A mi familia en general por ser mi razón de vivir.

Adonis Vázquez Campos.



Resumen

El proceso Extensión Cultural 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 formación de profesionales integrales, al fortalecimiento de los valores de la universidad cubana y a la promoción del desarrollo de la cultura para la comunidad interuniversitaria y extrauniversitaria.

El objetivo de la presente investigación consiste en el desarrollo de un sistema web para la gestión de información de los principales procesos de la Dirección de Extensión Cultural de la UCI. A partir del diagnóstico de un conjunto de indicadores que inciden en el óptimo funcionamiento de los procesos de gestión de información del Movimiento de Artistas Aficionados (MAA), se elaboró un marco teórico y el estudio del estado del arte donde se puso de manifiesto la necesidad de desarrollar el sistema propuesto. El análisis y diseño del sistema permitió la identificación de los requisitos para su posterior implementación.

La realización de las pruebas de software permitieron detectar y corregir los errores no identificados durante la implementación, posibilitando cumplir con las especificaciones requeridas y la validación del sistema implementado, las mismas arrojaron resultados satisfactorios demostrando la calidad del sistema desarrollado.

Como resultado se obtiene un sistema web para la gestión de la información de los procesos de la Dirección de Extensión Cultural de la UCI, cuyas funcionalidades responden a las necesidades del problema que se plantea, y a la vez permite un mejor desempeño del proceso.

Palabras clave: extensión cultural, gestión, sistema web.



Índice de contenidos

Datos de contacto.....	III
Agradecimientos	IV
Dedicatoria	V
Resumen	VI
Índice de contenidos	VII
Introducción	1
Fundamentación Teórica	7
Introducción.....	7
1.1 Conceptos asociados al dominio del problema.....	7
1.1.1 Extensión Universitaria:	7
1.1.2 Dirección de Extensión Cultural.	8
1.1.3 Afición.....	9
1.2 Procesos extensionistas culturales en la UCI.....	9
1.2.1 Movimiento de Artistas Aficionados.	9
1.2.2 Actividades culturales	9
1.2.3 Reservación.....	10
1.2.4 Patrimonio cultural de la UCI	10
1.2.5 Proyectos Socioculturales Comunitarios (PSC).	10
1.3 Las Tecnologías de la Información y las Comunicaciones en la gestión universitaria.	11
1.3.1 Portal web.....	12
1.3.2 Sistema Web	13
1.3.3 Sistemas de Gestión de Contenidos (CMS).....	13
1.4 Estado del arte.....	14



1.4.1	Portales de instituciones internacionales.	15
1.4.2	Portales de instituciones nacionales.	17
1.4.3	Valoración del estudio del estado del arte.....	18
1.5	Metodologías de desarrollo.	18
1.5.1	Selección de la metodología a utilizar.	19
1.5.2	Valoración de la metodología.....	21
1.6	Herramientas, Técnicas utilizadas y Lenguajes de Desarrollo.	21
1.6.1	Plataformas de desarrollo.	22
1.6.2	Selección del CMS a utilizar.	24
1.6.3	Sistemas de gestores de bases de datos.....	24
1.6.4	Selección del sistema gestor de bases de datos a utilizar.	26
1.6.5	Servidor Web.	26
1.7	Lenguajes de programación utilizados.	27
1.7.1	Lenguaje del lado del servidor.	27
1.7.2	Lenguajes del lado del cliente.....	28
	Capítulo 2: Solución Propuesta.....	31
	Introducción.	31
2.1	Propuesta de solución.....	31
2.2	Sistemas de control de acceso.....	32
2.2.1	RBAC.	32
2.2.2	Roles de usuario.....	33
2.3	Fases de SXP:	34
2.4	Especificación de los requisitos de software.....	35
2.4.1	Requisitos funcionales:	35
2.4.2	Requisitos no funcionales:	41
2.5	Historias de usuario.....	43



2.6	Plan de iteraciones.....	45
2.6.1	Descripción de las iteraciones.	45
2.7	Plan de duración de las iteraciones.....	45
2.8	Tareas de ingeniería.	47
2.9	Arquitectura del sistema.....	49
2.9.1	Patrón arquitectónico modular	49
2.9.2	Programación orientada a objetos en Drupal.....	50
2.10	Patrones de diseño.	52
	Conclusiones del capítulo	54
	Capítulo 3: Implementación y Prueba	55
	Introducción	55
3.1	Estándares de codificación utilizados.....	55
3.1.1	Identación.....	55
3.1.2	Etiquetas de apertura y cierre de PHP.....	56
3.1.3	Operadores	57
3.1.4	Uso de comillas	57
3.1.5	Uso de punto y coma (;) en código PHP	57
3.2	Diseño del sistema.....	58
3.3	Módulos de Drupal.....	58
3.4	Modelo de implementación.....	59
3.5	Fase de Implementación.	59
3.6	Fase de pruebas.	59
3.6.1	Pruebas unitarias en la metodología SXP.....	60
3.6.2	Pruebas de aceptación en la metodología SXP.....	60
3.6.3	Resumen de pruebas.....	62
3.7	Impacto de la solución propuesta.....	62



Conclusiones del capítulo.....	63
Conclusiones Generales.....	64
Recomendaciones.....	65
Referencias bibliográficas.....	66
Bibliografía.....	68
Anexos.....	71




Introducción

Con el creciente desarrollo de las tecnologías de la informática y las comunicaciones se ha facilitado el proceso de intercambio, gestión y publicación de la información, lo que hace indispensable la utilización de las tecnologías para hacer viable dicho proceso. Ciertamente para poder brindar una mayor cantidad de información en un mismo lugar, surgen en la década de los 90 los portales webs como punto de partida para la conexión a internet, bajo pilares fundamentales como: la información, tiene que ser gestionable bajo buscadores, directorios, noticias, catálogos o servicios, a su vez la participación de los usuarios, debe ser mediante chats, foros debates o e-mail. Con el fin de permitir que los usuarios interactuaran con los datos de forma personalizada y como tarea específica, surgen en la década de los 80 los sistemas webs brindando una mayor comodidad a la hora de gestionar la información. (Domínguez Fernández & Llorente, 2009).

En los últimos tiempos Cuba ha estado sometida a profundos y considerables cambios en busca de un desarrollo paulatino de cada uno de sus sistemas incluyendo el educacional; la batalla de ideas ha jugado un papel preponderante en la creación de nuevos programas con vistas a incrementar la calidad de vida de cada uno de sus ciudadanos, al permitir la creación de varios proyectos educativos y sociales.

Las universidades en Cuba constituyen pilares fundamentales en la creación de nuevos profesionales comprometidos con la Revolución, el entorno social y con un alto desarrollo socio-político y humanista. La Extensión Universitaria como proceso sustantivo para el desarrollo integral de los estudiantes, se plantea desde diferentes ámbitos o manifestaciones como el deporte, la cultura, la comunicación y la recreación sana, logrando un nivel de integralidad en el egresado, al igual que una mayor preparación para su desempeño profesional. La Extensión Cultural como subproceso en la relación dialéctica entre la universidad y la sociedad, posee un peso significativo en la búsqueda del perfeccionamiento de la Educación Superior Cubana, manifestándose desde dos aristas: una administrativa que opera desde las funciones de la dirección, cátedras de extensión, o departamentos de promoción y cultura; y la otra a partir de la dinámica de la creación, desarrollo y promoción sociocultural.

Por iniciativa del Líder de la Revolución Fidel Castro Ruz se crea al calor de la batalla de ideas la Universidad de las Ciencias Informáticas (UCI), con el objetivo de producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como



modelo de formación. Además sirve de soporte a la industria cubana de la informática (MisiónUCI, 2014). La UCI cuenta con un plan de estudio creado de acuerdo con la necesidad del país (210/07), de formar ingenieros altamente calificados en la rama de la informática y comprometidos con su Patria (MisiónUCI, 2014), el cual va de la mano de una política extensionista que tiene su punto culminante en la celebración de los Juegos Deportivos “Mella” y los Festivales de Artistas Aficionados (FAA).

El Programa Nacional de Extensión Universitaria (PRONAC-EU) de 2004, documento rector del proceso a nivel de país, plantea como objetivo estratégico “Desarrollar la extensión universitaria, transformándola a partir de asumirla como proceso orientado a la labor educativa, que promueva y eleve la cultura general integral de la comunidad universitaria y su entorno social” (Gonzalez Gil, 2004). Una de las funciones de la Dirección de Extensión Cultural (DEC) de la UCI es precisamente atender el Movimiento de Artistas Aficionados (MAA) de la FEU, movimiento este muy particular e identativo que cuenta con resultados importantes a nivel nacional. La DEC lleva a cabo todo el proceso de gestión de información del MAA, además de realizar la dirección artística de todas las actividades que se realizan en la Universidad. Es en esta Dirección donde se ofrecen ofertas culturales a los teatros, de tal manera la comunidad universitaria tiene la posibilidad de reservar entradas para dichos espacios.

No obstante se ha evidenciado con insatisfacciones por parte del personal que recibe los servicios extensionistas, una descentralización de la información generada en este proceso; debido a diferentes factores, dentro de los cuales se destacan:

✓ **Reservación y ventas de ofertas culturales (OC) :**

- Constantes colas a la hora tanto de reservar como para efectuar la compra de las OC.
- Limitada cantidad de papel, que se utiliza para la inscripción de los clientes.

✓ **Gestión de actividades culturales:**

- Falta de comunicación entre los directores artísticos y los colaboradores de las actividades.
- Engorroso trabajo de creación de nuevos guiones, debido a la gran cantidad de información que se genera.

✓ **Control de las actividades de los artistas aficionados:**



- El uso excesivo de la memoria conlleva a un defectuoso proceso de control de información de los artistas aficionados en las actividades, lo que tributa a una mala selección de los más destacados en el año.
 - No existencia de una base de datos, donde se tenga a todos los artistas que pertenecen al MAA.
 - Planillas con datos repetidos de los artistas, derivado de la elaboración manual de cada una de ellas.
 - Deterioro de algunas planillas producto del almacenamiento prolongado y gran cúmulo de las mismas.
- ✓ **Publicación de obras:**
- La galería tiene poco alcance, y no todas las personas pueden pasar a ver las exposiciones.
- ✓ **Patrimonio cultural de la UCI:**
- Desconocimiento en cuanto a:
 - Que obras conforman el catálogo del patrimonio cultural de la universidad.
 - Autores de las mismas.
 - Ubicación dentro de la universidad.
 - Descripción de la obra y de su autor.
- ✓ **Departamento de promoción:**
- Pobre divulgación de las actividades culturales y ofertas recreativas.

Todos los problemas anteriormente planteados, tributan a la falta de credibilidad por parte de la entidad, la pérdida de información relacionada con el MAA y la pobre visibilidad del mismo dentro de la comunidad universitaria.

La presente investigación forma parte y se encuentra respaldada por la **Dirección de Extensión Cultural** (DEC) centrada en la tramitación de una urgente respuesta que satisfaga la creciente situación que afronta, con el objetivo de viabilizar el trabajo que se realiza.

A partir de todo lo anteriormente planteado y la situación problemática descrita se propone en la siguiente investigación el siguiente **problema a resolver**:

¿Cómo facilitar la gestión de la información de los principales procesos de la Dirección de Extensión Cultural de la Universidad de las Ciencias Informáticas?



Por lo que se hace urgente llevar a cabo la misión de solucionar el problema anterior planteando como **objetivo general**: Desarrollar un sistema web que gestione la información de los principales procesos de la DEC en la Universidad de las Ciencias Informáticas.

Dicha investigación propone como **objeto de estudio**: Los procesos de gestión de la información en la extensión universitaria.

Definiendo como **campo de acción**: Los procesos de gestión de la información en la extensión universitaria en la UCI.


Como **posibles resultados** se espera: Un sistema web para la gestión de la información de los principales procesos de la DEC en la UCI, cuyas funcionalidades respondan a las necesidades del problema que se plantea, y a la vez permita un mejor desempeño del proceso.

Para darle cumplimiento al objetivo general planteado se responderán las siguientes **preguntas científicas**:

- ✓ ¿Qué elementos teóricos sustentan el desarrollo de un sistema web para la gestión de los principales procesos de la Dirección de Extensión Cultural de la UCI?
- ✓ ¿Qué aspectos deben tenerse en cuenta para realizar el análisis y el diseño del sistema web para la Dirección de Extensión Cultural de la UCI, de manera que se faciliten los procesos de la misma?
- ✓ ¿Cómo implementar, a partir del análisis y diseño realizado, el sistema web para la Dirección de Extensión Cultural de la UCI?
- ✓ ¿Qué resultados se obtienen al realizarle las pruebas al sistema web para la Dirección de Extensión Cultural de la UCI?

Las **tareas de investigación** a desarrollar para dar respuesta a lo anterior son:

- ✓ Realización de un estudio sobre las tendencias en el desarrollo de portales web para la gestión de procesos extensionistas.
- ✓ Realización del análisis y el diseño del sistema web para la gestión de la información de los principales procesos de la DEC de la UCI.
- ✓ Implementación del sistema web para la gestión de la información de los



principales procesos de la DEC de la UCI.

- ✓ Realización de las pruebas al sistema web implementado.

Para lograr el pleno y óptimo desarrollo de la investigación se hace indispensable el empleo y uso de **métodos científicos**, específicamente los que se evidencian a continuación:

Métodos teóricos: permiten estudiar las características del objeto de investigación que no se observan directamente, además crean las condiciones para ir más allá de las características fenomenológicas y superficiales de la realidad. Se aplican en calidad de enfoque general como estrategia. Los que se deben utilizar son:

- ✓ Analítico-sintético: se realizó el estudio de la literatura especializada relacionada con el tema y la exploración de resultados de investigaciones afines, lo que permitió adoptar posiciones teóricas relacionadas con el objeto de investigación.
- ✓ Revisión documental: se realizó una revisión de los documentos que sustentan el proceso de EC en la UCI, para una mayor preparación y desarrollo de la investigación.

Métodos empíricos: estos métodos explican las características fenomenológicas del objeto, con ellos es posible representar un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional. Se aplican como procedimiento en el proceso de investigación. De los métodos empíricos existentes se utilizarán:

- ✓ Entrevistas no estructuradas o libres: se le realizaron a especialistas de la dirección de extensión, así como a todo el personal que accede al centro cultural de manera continua, con el objetivo de obtener información relacionada con la aplicación y los diferentes conceptos a utilizar, lo que contribuye al perfeccionamiento de la propuesta realizada.
- ✓ La observación: mediante el cual se valoraron los avances realizados en el estudio de las diferentes herramientas y tecnologías a usar, además se conoció de qué forma las personas buscan en estos sistemas y cómo funcionan.

Este documento presenta la siguiente estructura:

Capítulo I Fundamentación teórica: se realiza un análisis detallado del marco conceptual, donde se trata los principales conceptos relacionados con el tema en cuestión. Se



analizan sistemas similares, además del estado del arte de las tecnologías y herramientas a utilizar en el desarrollo de la aplicación.

Capítulo II Solución Propuesta: se describe la propuesta del sistema, se detallan los artefactos generados en las fases Planificación-Definición y Desarrollo, prestando especial atención a las historias de usuario, plan de iteración y tareas de ingeniería. Además, se exponen los aspectos relacionados con el diseño del sistema donde se manifiestan los artefactos propuestos en esta fase, además se argumentan algunas de las terminologías que se utilizan.

Capítulo III Implementación y pruebas: es donde se orienta a la implementación del sistema de gestión y a los componentes y módulos utilizados. Además, se describen las tareas relacionadas en la fase de implementación. Se exponen los aspectos relacionados con la fase de pruebas. Se detallan los artefactos propuestos en esta fase y se llevan a cabo las pruebas de aceptación solicitadas por el cliente.



Fundamentación Teórica

Introducción

El objetivo de este capítulo es profundizar en el análisis del estado del arte del tema propuesto, así como también plantear el resultado del estudio de las posibles herramientas y tecnologías a utilizar para el desarrollo de la aplicación en cuestión, el cual favorece el desarrollo de la investigación y permite que el cumplimiento de los objetivos generales de la investigación y la aplicación en sí, sean los más óptimos posibles.

1.1 Conceptos asociados al dominio del problema.

La Extensión Universitaria en las Universidades

La Extensión Universitaria (EU) juega un papel importante dentro de las universidades, pues posee como misión fundamental el encargo social de preservar, desarrollar y promover la cultura, permitiendo desarrollar un sistema de comunicación interna y externa que propicia el diálogo, potencia la participación, posibilita la difusión y divulgación de la cultura y el quehacer universitario y social, contribuyendo así a la elevación del nivel cultural de la comunidad, su residencia y su entorno social. Dentro de la UCI la Extensión Universitaria (EU) posee varias aristas que, para el desarrollo integral de los estudiantes y trabajadores se manifiesta en deporte, cultura, comunicación, proyectos comunitarios y socioculturales. Ahora bien, se puede definir que:

1.1.1 Extensión Universitaria:

Ámbito Internacional

Universidad "Francisco Gavidia" de El Salvador.

Según estudios realizados en la Universidad Francisco Gavidia (UFG) de El Salvador la Extensión Universitaria es por definición la dimensión valorativa de la vinculación social de la universidad, que se concretiza con la participación activa, responsable y auto determinada de cada uno de los miembros, por tanto se convierte en un **"programa integral de docencia-investigación y proyección social, a través del conocimiento de la realidad para crear una conciencia que contribuya al desarrollo nacional"**.



Intelectual Brasileño Frey Betto.

Durante la tercera jornada del 8vo. Congreso Internacional Universidad 2012, este intelectual brasileño quien dictara una conferencia magistral titula **“Extensión universitaria y educación popular”**, definió la extensión universitaria como un reto concluyendo además que **“debe llevar la universidad a la realidad política, involucrarse en los procesos sociales de los pueblos, y extraer de ellos sus vivencias y experiencias espirituales, religiosas, lúdicas”**.

Cuba

Citando el PROGNAC-EU de 2004, se está en presencia de un proceso universitario formativo y lo define como **“el proceso que tiene como objetivo promover la cultura en la comunidad interuniversitaria y extrauniversitaria, para contribuir a su desarrollo cultural”**.

Concepto ofrecido por la Dra. María Elena del Huerto Marimón¹.

"La Extensión Universitaria es una de las vías para desarrollar el proceso de comunicación cultural en un sentido amplio, facilita el vínculo Universidad-Sociedad a través de sus diferentes formas de manifestación, presta servicios a la comunidad universitaria y a la población en general para dar respuesta a las necesidades de capacitación y contribuir al desarrollo cultural integral." (Del Huerto Marimón, y otros, 2010).

1.1.2 Dirección de Extensión Cultural.

Es el primer órgano rector del proceso integrador y dinamizador del flujo cultural y la comunicación entre la universidad y la sociedad así como también entre cada uno de sus miembros.

La DEC en la UCI dentro de sus líneas de trabajo cuenta con la creación de proyectos socioculturales comunitarios con el fin de estrechar cada vez más los lazos de integración de la universidad con la sociedad, así como también cuenta con un MAA que permite estimular el desarrollo de la extensión, además de impulsar la creación y el promoción del arte como un medio en el que estudiante se sienta y comporte como universitario en el

¹ Doctora en Ciencias de la Educación. Profesora Titular. Asesora Oficina de Grados Científicos. Vicerrectoría de Postgrado e Investigación. Universidad de Ciencias Médicas de la Habana. CUBA.



plano de su disfrute cultural e ideológico, con un alto nivel de sensibilidad y de identidad con su centro.

Es bueno acotar que la terminología extensión cultural es propia de la UCI, nombre este dado por la MsC. María de la Caridad Ramírez Santos, para evitar confusiones en la documentación de la Vice-rectoría de extensión.

1.1.3 Afición.

Afición deriva de la lengua francesa, a su vez derivada de la raíz de la palabra en latín, la cual significa "amar a" o "el amador de", lo que avala que un amateur puede ser tan hábil como un profesional, sin embargo su motivación es el amor o la pasión por una cierta actividad y no tiene el fin de ganar dinero por realizarla.

Una afición (también llamado hobby, pasatiempo, distracción) es una actividad cuyo valor reside en el entretenimiento de aquél que lo ejecuta, que algunas veces no busca una finalidad productiva concreta y se realiza en forma habitual.

1.2 Procesos extensionistas culturales en la UCI.

1.2.1 Movimiento de Artistas Aficionados.

La UCI posee un MAA integrado por estudiantes y profesores que de conjunto son el agente de una alegría que prende en toda la comunidad, así como también depositarios de una responsabilidad que solo se evidencia en las cotas de calidad que se proponen alcanzar con su arte. Los Festivales de Artistas Aficionados (FAA), tanto de estudiantes como de trabajadores, constituyen el momento cumbre de este movimiento, que se realiza de carácter anual en eventos a nivel de facultad, universidad, municipales y provinciales, así como nacional cada dos años.

El movimiento es reconocido por el Ministerio de Cultura, como uno de los de mayor nivel y calidad en el país dentro de la FEU, avalado por su sistematicidad y proyección social.

Para un artista aficionado poder participar en esta actividad, debe presentar una planilla de inscripción con un conjunto de datos del mismo, como: nombre y apellidos, carnet de identidad, número de solapín, facultad, género y año (en el caso de los graduados debe especificar el año de graduado). Planilla que es gestionada en el Departamento de Gestión Extensionista (DGE) de la DEC.

1.2.2 Actividades culturales



La DEC lleva a cabo la gestión de un plan de actividades extracurriculares, como son: festivales, peñas, actos políticos, recibimiento a delegaciones extranjeras y cubanas, cafés literarios, presentaciones de libros y otras actividades que surgen sin previa planificación. Dichas actividades son gestionadas a diferentes niveles, por ejemplo: a cada actividad se le otorga un nivel, lo que influirá en el tipo de local que se escogerá para hacerla y los artistas que participan en ella.

La DEC estableció 3 niveles fundamentales: un primer nivel para las actividades protocolares (visitas de empresarios extranjeros, visitas ministeriales), un segundo nivel para las actividades centrales (acto político-cultural en las diferentes plazas, marchas), y un tercer nivel que comprende todas aquellas actividades de pequeña envergadura que no necesiten mucho despliegue técnico, es decir peñas de facultad, matutinos, parada de becas, entre otras.

1.2.3 Reservación

En la DEC, existe un convenio con varios teatros de la capital, para tener acceso a un conjunto de entradas para conciertos, festivales nacionales e internacionales, puesta en escena de obras teatrales y gala de premiaciones. El Departamento de Promoción (DP), gestiona este proceso de la siguiente forma: se publica la oferta cultural en la intranet de la universidad, específicamente en el apartado extensionista, luego el interesado debe dirigirse al DP para hacer la reservación y una segunda vez para comprar la oferta, superando innumerables colas y sobre todo en horario laboral.

1.2.4 Patrimonio cultural de la UCI

La UCI cuenta con un rico patrimonio cultural con obras de artistas importantes de nuestro país como Kcho, Flora Font, Wifredo Lam y extranjeros como Niemeyer y Secsuko Ono. Pocas son las personas que conocen en detalles a quien pertenece cada una de estas obras, donde están enclavadas dentro de la universidad y que significan. Toda la documentación referente a la historia de las obras y sus autores se encuentra archivada en el DGE de la DEC, en formato duro.

1.2.5 Proyectos Socioculturales Comunitarios (PSC).

El PRONAC-EU lo define como un proceso encaminado al estudio y evaluación de las posibles soluciones de los problemas de la comunidad en general que pueden



promoverse desde el contacto directo con los pobladores, lo que constituye un elemento esencial en su formación como profesionales revolucionarios.

Los PSC pueden ser de varios tipos como:

- ✓ **Participativo:** involucra a estudiantes, docentes, tutores, trabajadores, universidad y entorno social desde una postura activa y de transformación de la realidad.
- ✓ **Dialógico:** tributa al establecimiento de una realidad dialogal y tiende al intercambio de propuestas entre la universidad y la sociedad.
- ✓ **Creativo:** genera acciones innovadoras y trascendentes.
- ✓ **Contextualizado:** adaptable al medio en que se contextualiza y evoluciona dinámicamente en función del cambio que experimenten las condiciones en las que se desenvuelve.

Algunos de los proyectos socioculturales comunitarios creados en la UCI son:

Proyecto sociocultural comunitario Mala Testa.

Surge en las Alturas de la Lisa en el año 2013 con la finalidad de construir un espacio sociocultural para la comunidad, donde se opere idóneamente sobre la realidad social y que logre la integración de los jóvenes a través de las distintas formas de participación. Mediante la ejecución de una serie de acciones comunitarias vinculadas con el arte, el proyecto desea erradicar aquellas prácticas sociales negativas, reproducidas y naturalizadas socialmente, que afectan la calidad de vida de la población, así como convertirse en eslabón fundamental de la vida sociocultural de la población de 18 a 35 años.

Visitas de la alegría

Surgen como actividades extensionistas, propuestas por las facultades, con el objetivo fundamental de mejorar la calidad de vida de ancianos, niños sin amparo filial y pacientes hospitalizados. Por la magnitud que fueron alcanzando se convirtieron en proyectos socioculturales, mejores estructurados y con mayor cantidad de acciones.

1.3 Las Tecnologías de la Información y las Comunicaciones en la gestión universitaria.



El creciente desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC), ha hecho que se reemplacen los antiguos métodos de intercambio, gestión y publicación de información y forma de trabajo manual por sistemas digitales o sistemas web (Ecured, 2013). Estos brindan la posibilidad de realizar una gran cantidad de procesos y acciones de una manera rápida y sencilla, además constituyen un medio viable para que los trabajadores puedan hacer sus propuestas y comentarios acerca de diversos temas, participar en foros y encuestas, con el objetivo de intercambiar ideas e información con personas de cualquier lugar. Podemos concluir que las TIC son aquellas herramientas computacionales e informáticas que procesan, almacenan, sintetizan, recuperan y presentan información representada de la más variada forma.

1.3.1 Portal web

Un portal web no es más que un punto de partida que ofrece al usuario el acceso de forma segura, unificada y sencilla, a una disímil cantidad de información integrada proveniente de orígenes diferentes, instituciones científicas, u otros portales web, así como recursos y servicios relacionados a un tema en específico. El cual incluye: foros, documentos, aplicaciones, buscadores, tiendas online, enlaces con otros sitios, galerías de imágenes y videos (García Gómez, Juan Carlos), (Informática Milenium, 1998_2006).

Clasificación de los Portales Web.

- ✓ **Portales Horizontales o Genéricos:** también conocidos como mega portales, cuyo contenido está dirigido a todo tipo de usuario, en el que se brinda cierta cantidad de información de forma estructurada, sin obligar a los usuarios a navegar por muchos sitios para obtenerla, en otras palabras diseñado para usuarios con poca experiencia en el internet. Incluyen servicios como: Chats, Comunidades virtuales, e-mail, Foros de discusión, Noticias, Información, etc. (García Gómez, Juan Carlos)

Ejemplos: www.AltaVista.com, www.Lycos.com, www.Yahoo.com,
www.MSN.com.

- ✓ **Portales verticales o especializados:** proveen de información y servicios a un sector en particular, centrados en un tema específico, como música, deportes, finanzas, arte, comida, etc. acotar que en muchos de los casos el acceso no es gratuito y se trata de Portales Corporativos que dentro de una Intranet Corporativa,



proporciona a sus miembros de información interna y externa relacionada con la propia entidad. (García Gómez, Juan Carlos)

Ejemplos: www.softonic.com: dedicado específicamente al desarrollo del software, los usuarios pueden descargar programas y a su vez dejar comentarios sobre los mismos.

www.gobiernoelectronico.org: este portal está dedicado al tema de los gobiernos electrónicos en América Latina.

- ✓ **Portales diagonales:** también conocidos como híbridos, están formados por el Portal Horizontal y el Vertical, abordando temas variados, de naturalezas diferentes, así como profundidad y alcance, son utilizados mayormente en redes sociales o aplicaciones generalistas dirigidas a un público muy concreto. (García Gómez, Juan Carlos)

Ejemplos: www.Facebook.com, www.Linkedin.com, www.YouTube.com.

- ✓ **Portales móviles:** son aquellos portales que permiten la conexión tanto desde internet, como de un teléfono móvil o PDA. Este es similar a un sitio web normal, pero diseñado para pantallas pequeñas, de tal forma que las imágenes y los contenidos se optimizan para que se ajusten a las pantallas de dispositivos móviles. Cada vez más se abren caminos a niveles empresariales para realizar negocios ofreciendo productos o servicios a los usuarios. (García Gómez, Juan Carlos)

Ejemplos: www.miempresa.com, www.mfacebook.com, así como todos los portales que estén diseñados para ser compatibles con las tecnologías móviles como *Android*, *iOS (iPhone)*, *BlackBerry OS*, etc.

1.3.2 Sistema Web

Los sistemas web surgen por la necesidad de realizar tareas específicas sobre diferentes datos, así como para interactuar de manera directa con ellos. Por lo que se puede definir como un conjunto de programas para ayudar al usuario de un ordenador a procesar una tarea específica (Oxford, 2013). Según un estudio realizado por Walter Ulises Ayllapan, investigador para el desarrollo de Drupal y Joomla se trata de un sistema computacional que se utilice para obtener, almacenar, manipular, administrar, controlar, procesar, transmitir o recibir datos, para satisfacer una necesidad de información.

1.3.3 Sistemas de Gestión de Contenidos (CMS).



En la constante evolución de las herramientas para la creación de sitios webs se encuentran los CMS, los cuales facilitan la producción, administración y actualización de manera simple y dinámica y el mantenimiento de los sitios web, además separan el contenido del diseño. Juan Julián Mérelo Guervós², en su página personal ofrece una adecuada definición de CMS. Un CMS se trata generalmente de un conjunto de herramientas, apoyado habitualmente por una base de datos, que consisten en una serie de programas en un servidor web y opcionalmente, una serie de programas cliente que permitan acceder fácilmente a esos programas en el servidor (Merelo Guervós, 2005).

✓ **Ventajas de los CMS.**

- **Sencilla administración:** la información se introduce a través de formularios mediante el uso de un navegador web. La administración completa se realiza desde una única aplicación.
- **Control de acceso:** permite gestionar los diferentes permisos a cada área del sitio aplicados a grupos o individuos.
- **Cambiar el diseño del sitio:** es muy fácil de cambiar, gracias al sistema de plantillas que utiliza, las cuales se instalan y luego se seleccionan desde la interfaz administrativa.
- **Administrar usuarios:** desde la interfaz administrativa también se controla completamente a los usuarios del sitio, pudiendo especificar niveles jerárquicos, editar perfiles, censurar, dar permisos de publicación.
- **Administrar módulos y componentes:** tiene la posibilidad de instalar, desinstalar y administrar componentes y módulos, por ejemplo galerías de fotos, sistemas de votaciones.

Según la compañía *iMajestic Internet Marketing* los CMS más usados en la actualidad en el desarrollo de aplicaciones y portales web son: **Drupal** y **Joomla**.

1.4 Estado del arte.

Alrededor del mundo son muchas las instituciones que utilizan sistemas, así como portales que, de manera automatizada permitan la publicación, gestión y promoción de la información que generan. Algunos de estos sistemas cuentan con políticas privativas

² Juan Julián Mérelo Guervós, licenciado en Ciencias Físicas por la Universidad de Granada en 1988 y doctorado por la misma universidad en 1994. Actualmente es Profesor Titular de la Universidad, asignado al Depto. de Arquitectura y tecnología de Computadores, e imparte docencia en la ETS de Ingeniería Informática.



empleadas para proteger el mal uso de su información, la violación de los derechos de sus autores y cuidar el prestigio de dicha institución.

1.4.1 Portales de instituciones internacionales.

Portal de extensión universitaria de la Universidad de la Laguna (ULL).

La ULL se encuentra en San Cristóbal de La Laguna, Santa Cruz de Tenerife, España. La web permite una navegación intuitiva, interfaz clara, lo cual evita la sobrecarga de elementos con el fin de no confundir a los usuarios. Brinda además la posibilidad de filtrar la información que necesita el cliente mediante un buscador; posee un módulo de cultura y extensión, y este a su vez incluye varios sub-módulos donde se publica información referente a cursos y talleres, actividades culturales, colectivos culturales, aulas y cátedras culturales y premios.

Portal de la Vicerrectoría de Extensión Universitaria de la Universidad de Extremadura (UNEX).

La UNEX se encuentra en la provincia de Badajoz en España. El objetivo fundamental del portal UNEX es utilizar la web como principal canal de información útil, actualizada para toda la comunidad universitaria extremeña. Se ha realizado en la nueva versión una reestructuración de la información con la que se ha pretendido:

- ✓ Segmentar el contenido de la web en función de los tres grandes perfiles de usuario: estudiantes, personal trabajador y empresas. Clasificar la navegación en función de sus necesidades.
- ✓ Separar la navegación en dos grandes bloques: usuarios expertos, que utilizan la web como herramienta recurrente de información o de trabajo, otorgándoles un tipo de navegación más avanzada; y usuarios inexpertos, que acceden a la web por primera vez, merecedores de un sistema de navegación más intuitivo.

La Vicerrectoría de extensión universitaria, está dividida en 7 departamentos que para el portal se traducen en módulos como son:

- ✓ Servicios de publicaciones.
- ✓ Secretariado de actividades culturales.
- ✓ Servicio de acción social y asistencial.
- ✓ Servicio de asistencia religiosa.



- ✓ Residencia V centenario de jarandilla de la vera.
- ✓ Instituto de lenguas modernas.
- ✓ Universidad de mayores.

Cada uno de estos módulos, se dedican a la publicación de información referente a cursos de extensión, datos de los directivos de las distintas áreas de trabajo, así como de las funciones de cada una de las aulas de danza, música, teatro, literatura, cine y video y fotografía. Decir que esta web está desarrollada con lenguaje PHP. Para el diseño se utilizo HTML 5, y recomienda como navegador web versiones superiores a la v.4 de Mozilla Firefox.

Portal de la Universidad de Harvard (HU)

La Universidad de Harvard es una universidad privada ubicada en Cambridge, Massachusetts, Estados Unidos. Forma parte de la Ivy League ³ y es la institución de enseñanza superior más antigua de los Estados Unidos. Fundada el 13 de marzo de 1639 por un sacerdote llamado John Harvard. La institución cuenta con una facultad de Artes y Ciencias con la Escuela de Ingeniería y Ciencias Aplicadas, compuesta de: *Harvard College, The Graduate School of Arts and Sciences y The Harvard Division of Continuing Education, que incluye la Harvard Extension School y la Escuela de Verano de Harvard.* (2013)

Algunos ejemplos de condiciones de uso

Acuerdo del usuario

De la propiedad intelectual: Harvard se reserva todos los derechos, incluidos los derechos de autor, en el contenido, sujeto a la limitación de que los derechos de autor y otros derechos de propiedad pueden ser ocupados por personas o entidades distintas, o además de, la Universidad de Harvard.

Licencia limitada general: usted no va a reproducir, mostrar, distribuir, ejecutar, o hacer cualquier otro uso de HU Contenido del portal, sin obtener primero el permiso específico de la Universidad de Harvard.

El uso del contenido de tercero: Harvard no garantiza que cualquier uso de contenido de terceros no infringirá los derechos de los propietarios de contenido.

³ **Ivy League** (*Liga Ivy* o *Liga de la Hiedra*) es una conferencia deportiva de la NCAA de ocho universidades privadas del noreste de los Estados Unidos.



Indemnización: usted se compromete a indemnizar y mantener indemne a la Universidad de Harvard y sus filiales, agentes, oficiales, directores, y empleados de y contra cualquier y toda responsabilidad, pérdidas, reclamaciones, daños, costos y / o acciones (incluyendo honorarios de abogados) que surjan del uso del contenido, el sitio web de la HU.

Declaración de privacidad: el sitio web de la HU contiene enlaces a otros sitios web que funcionan independientemente de la red de la Universidad de Harvard, y en algunos sitios fuera de la "harvard.edu" de dominio. La HU no se hace responsable por las prácticas de privacidad o el contenido de dichos sitios web. (2013)

Portal de la Universidad de Oviedo.

La Web institucional de la Universidad de Oviedo se desarrolló sobre un CMS creado por el administrador del portal. Se trata de ZList que funciona sobre Zope⁴, está hecha acorde a las necesidades de la universidad. La idea del CMS ZList está inspirada en tres condiciones:

- Posea una interfaz estándar para todas las páginas y pueda ser modificada por cualquier usuario.
- La información debe estar almacenada en un mismo lugar y la información pueda ser publicada en distintas páginas con distintos formatos.
- Debe existir un modelo por defecto listo para usarse sin intervención de un desarrollador web.

La extensión universitaria en este portal se aborda de una manera particular, debido a que no existe un módulo únicamente destinado a los procesos extensionistas, se publica la información por separado en distintos módulos.

1.4.2 Portales de instituciones nacionales.

Además existen otros centros en Cuba los cuales cuentan con portales que a la vez gestionan gran cantidad de información como son:

Portal Digital de la Universidad de las Ciencias Informáticas.

La UCI posee un portal digital para internet mediante el cual podemos conocer más sobre la historia y el quehacer universitario de la institución. Dentro de los módulos que presenta

⁴ **Zope** es un entorno de desarrollo para la creación de sitios web dinámicos y/o aplicaciones web usando un servidor de aplicaciones web orientado al objeto, escrito en el lenguaje de programación *Python* (con algunos componentes escritos en lenguaje C para optimizar su rendimiento) de código abierto publicado bajo la licencia *Zope Public License*.



se encuentra Vida universitaria, el cual está dividido en cinco sub-módulos: cultura, deporte, comunicación, residencia y galerías, donde nos permite un acercamiento a las principales actividades y acciones que ofrece la universidad en esa rama.

Portal de la Facultad de Matemática y Computación de la universidad de la Habana

El portal de la universidad de la habana nos ofrece servicios de búsqueda de información, biblioteca, estadísticas de acceso. Para el desarrollo del mismo se utilizaron dos CMS: *Rainbow* y *DotNetNuke* versión 3. La extensión universitaria en el mismo es abordada desde la perspectiva de la publicación de información de 3 sub-módulos: cátedras honoríficas, grupos institucionales y proyecto comunitarios, así como la publicación de noticias relacionadas con diversos temas.

1.4.3 Valoración del estudio del estado del arte.

Luego de realizar el estudio de los sistemas, se arriba que el 100% de ellos se dedican a la publicación de la información de sus respectivas instituciones sobre algún tema de discusión, poseen buscadores de gran eficiencia, mientras que UNEX es capaz de ayudar a la hora de tomar decisiones y HARVARD resalta por su estructura organizativa a la hora de guardar información en el repositorio.

Luego de analizar las soluciones asociadas tanto al proceso de extensión como a la publicación de información para una comunidad en específico, la propuesta más acertada es la que brinda la UCI para la gestión universitaria, la cual cumple con el 80 % de las características más importantes que se requieren para dar solución al problema planteado, a continuación se presentan algunas de ellas:

- ✓ Interfaz amigable y con facilidad de uso.
- ✓ Sencillez en el diseño, característica que proporciona una interfaz amigable.
- ✓ Empleo de párrafos cortos, lo que facilita la lectura por parte del usuario.
- ✓ Tamaño justo de las fuentes, lo que permite la legibilidad del texto.

A lo cual se concluye que el sistema que se desarrolla como objetivo de la presente investigación es una alternativa ante la inexistencia de un centro cuya gestión informática de la información sea efectiva y accesible desde un entorno digital, el cual fomente el conocimiento de diferentes esferas de la cultura en la sociedad.

1.5 Metodologías de desarrollo.



Las metodologías son necesarias e importantes para obtener la calidad del producto final, se pueden clasificar en: **tradicionales** (pesadas o de peso pesado) **y ágiles** (Ligeras). Cuando el proyecto a realizar es demasiado grande se utilizan metodologías tradicionales, estas se centran en una mayor organización y una manera más detallada de cada una de las tareas a realizar, así como herramientas a utilizar. Sin embargo las metodologías Ágiles permiten la comunicación cara a cara entre los miembros del proyecto y los clientes, además ofrecen una buena solución para proyectos de corta duración donde los requisitos no se conocen con exactitud, pues están pensadas para trabajar con incertidumbre (Figuroa, Solís, & Cabrera, 2008). Por todo lo antes expuesto se determina la utilización de una metodología ágil, para el proceso de desarrollo de software del Sistema Web de la Dirección de Extensión Cultural de la Universidad de las Ciencias Informáticas.

Ejemplos de metodologías Tradicionales y Ágiles:

Tradicionales:

- RUP (Proceso Unificado de Desarrollo).
- Métrica_3.

Ágiles:

- XP – *extreme Programming*.
- Scrum.
- FDD – *Feature Driven Development*.
- SXP (Scrum-XP).

1.5.1 Selección de la metodología a utilizar.

A continuación se realiza una comparación entre los dos tipos de metodologías tradicionales y ágiles (Confección propia, 2014).

Metodologías tradicionales	Metodologías ágiles
El cliente interactúa con el equipo de desarrollo solo en reuniones.	El cliente es parte del equipo de desarrollo.
Más artefactos	Menos artefactos
Más roles	Pocos roles

Ideal para proyectos grandes.	Ideal para proyectos de corta duración y pequeños, es decir, donde el número de integrantes sea menor de 10.
-------------------------------	--

Tabla 1: Comparación de las Metodologías Tradicionales y Ágiles.

Por lo anteriormente planteado se valora dentro de las metodologías ágiles, la utilización de **SXP**:



FIGURA 1. METODOLOGÍA SXP.

La misma integra Scrum y XP, dos de las mejores metodologías ágiles de desarrollo de software que se utilizan en la actualidad. Además de ser una metodología definida por la UCI, está especialmente indicada para proyectos pequeños, que presentan un rápido cambio de requisitos donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo.

Permite el intercambio constante con el cliente y a su vez está dividida en cuatro fases, que son precisamente la base de la estructura del nuevo expediente de proyecto que brinda, estas son:

- Planificación-Definición: se establece la visión, se fijan las expectativas y se asegura el financiamiento del proyecto.
- Desarrollo: se realiza la implementación del proyecto.
- Entrega: es la puesta en marcha.
- Mantenimiento: aquí se lleva a cabo toda la parte de soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición



de las Historias de Usuario, diseño, implementación, pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, y existe una refactorización continua, lo que permite mejorar el diseño cada vez que se le añade una nueva funcionalidad (Romero, 2013). Con la utilización de SCRUM para la gestión, se logra una planificación y organización inigualable; mientras que XP respalda con sus prácticas todo el proceso de desarrollo, obteniéndose de esta forma un proceso de software completo. (Romero, 2013)

1.5.2 Valoración de la metodología.

Las metodologías proponen la documentación de los productos que en ocasiones suelen ser extensas, con el surgimiento de SXP nació la necesidad de buscar una nueva forma de documentar los productos que fueran desarrollándose de manera que respondieran al proceso de desarrollo de las metodologías ágiles. Es así como empezaron a utilizarse algunas de las planillas que proponen estas metodologías con el objetivo de que el proceso de documentación de software fluyera de manera rápida en los proyectos, tales como: concepción del sistema, que permitió resumir los aspectos más relevantes del sistema, historias de usuario, que permitió ratificar que en ellas se resumía todo lo referente a las especificaciones de los requerimientos y gestión de cambio, que brindó la recopilación de cada una de las mejoras realizadas a los productos.

Con la utilización de estas plantillas se alcanzaron resultados significativos, como son:

- La entrega en tiempo y forma de la documentación mínima y necesaria, de la mayoría de los proyectos que trabajan con la metodología SXP.
- Un proceso de documentación menos tedioso, y con resultados relevantes.
- Una gestión más sencilla y eficiente por todos los niveles de dirección de proyectos.

Por lo anteriormente planteado se puede concluir que con esta metodología se pretende dar solución a muchos de los problemas que afectan el proceso de documentación en los proyectos productivos, como son las documentaciones extensas, la duplicación de esfuerzos, así como el tiempo que se dedica a documentar, que en la mayoría de los casos es mayor que el dedicado a la producción.

1.6 Herramientas, Técnicas utilizadas y Lenguajes de Desarrollo.



En la realización del producto, se hizo necesaria la utilización de distintas herramientas, que permitieran el óptimo desarrollo del mismo. Es por ello que se realizó un análisis previo del estado de algunas tecnologías actuales y sus características.

1.6.1 Plataformas de desarrollo.

Drupal 7.26.

Sistema de gestión de contenidos (CMS) que se distribuye como software libre bajo licencia GNU/GPL (*General Public License o Licencia Pública General*). Está desarrollado completamente por lenguaje PHP, es flexible y adaptable, así como dispone de gran cantidad de módulos adicionales que permite que sea adecuado para la realización de muchos sitios Web (RODRÍGUEZ, 2012).

Ventajas de Drupal:

- Acceso rápido a la web.
- Permite crear sitios con contenido en varios idiomas.
- Permite crear sitios web con mucho contenido, como diarios digitales o portales multimedia.
- Drupal 7.26 cuenta con más de 2.500 módulos, los cuales brindan un gran alcance funcional.
- Integra un potente gestor de menús que permite disponer de múltiples y variados menús en el mismo sitio web.
- Los usuarios se pueden registrar e iniciar sesión de forma local o utilizando un sistema de autenticación externo. Para su uso en una intranet, Drupal se puede integrar con un servidor LDAP.
- Permite a los usuarios configurar su propio perfil en dependencia de los permisos otorgados por el administrador (Reyero, 2005).

Joomla

El CMS Joomla es un sistema de código abierto construido con PHP, bajo una licencia GPL, además utiliza una base de datos MySQL. Joomla es un sistema gestor de contenidos dinámicos (CMS o *Content Management System*) que permite crear sitios Web de alta interactividad, profesionalidad y eficiencia (Ferrer, 2007). Presenta una interfaz sencilla y amigable, cualquier persona puede administrar sus propios contenidos Web sin la necesidad de poseer conocimientos técnicos, sin saber lenguaje HTML y sin recurrir a



un web-master cada vez que hay que actualizar alguna información en un sitio Web. Entre sus características se encuentra: la versatilidad del sistema a través de plantillas, extensiones y adaptaciones; la posibilidad de hacer caché de páginas para mejorar el rendimiento; versiones imprimibles de páginas; indexado web; entre otras (Ayllapan).

Desventajas de Joomla:

- Deficiente planificación de la interfaz administrativa.
- Limitación en las opciones para personalizar rangos de usuarios.
- Dependencia excesiva del *JavaScript* en su panel de administración.
- Es un poco lento.

A continuación se enuncian algunas de las características a comparar entre los CMS Drupal y Joomla:

- **Última versión estable:** Se especifica la versión estable más reciente de cada uno de estos CMS a analizar.
- **Fecha de liberación de la última versión estable:** La fecha de actualización de la versión estable.
- **Seguridad:** Mecanismos implementados para auditar los accesos de los usuarios, integración de mecanismo de autenticación.
- **Arquitectura:** Arquitectura de desarrollo que presenta la herramienta.
- **Interoperabilidad:** Integración de APIs de servicios y de librerías externas.

Tabla 2. Comparativa entre los CMS Drupal y Joomla (Confección propia, 2014).

CMS	Drupal	Joomla
Última Versión Estable	7.26	3.0.2
Fecha de liberación de la última versión estable	7 de marzo de 2013	8 de noviembre de 2012
Seguridad	Sistema basado en roles, presenta un módulo para auditar los accesos de los usuarios, la autenticación es extensible a través de módulos.	No presenta mecanismos para registro de trazas de la actividad de los usuarios. La autenticación es extensible a través de <i>plugins</i> .



CMS	Drupal	Joomla
Arquitectura	Modular y extensible a través de la implementación de ganchos (<i>hooks</i>).	Extensible basada en <i>plugins</i> .
Interoperabilidad	Extensible a través de módulos, fácil integración con librerías externas de manera nativa.	Limitada a través de <i>plugins</i> , fácil integración con librerías externas de manera nativa.

1.6.2 Selección del CMS a utilizar.

Para el desarrollo de la propuesta de solución, se determina utilizar el CMS Drupal en su versión 7.26. La selección se justifica por las características presentes en Drupal, logrando que el desarrollo de la solución resulte rápido y satisfaga los requerimientos identificados:

- El sistema de permisos que define Drupal, brinda la posibilidad de delimitar el sistema de acceso a los campos y a la entidades de manera sencilla, utilizando roles para los usuarios que acceden al sistema (Peña, Rodríguez, Carlos, & Dr. Paredes García, Fernando).

1.6.3 Sistemas de gestores de bases de datos.

A continuación se hará énfasis en dos de los gestores de base de datos, con los cuales el CMS Drupal guarda más relación.

MySQL 5.5

MySQL es un sistema de gestión de base de datos relacional, lo que permite velocidad y flexibilidad, en el cual los datos están almacenados en tablas separadas enlazadas entre sí. La parte SQL de "MySQL" significa "Lenguaje Estructurado de Consulta" y fue creado por una empresa sueca que se llama MySQL AB. El software MySQL tiene licencia dual, pudiéndose usar de forma gratuita bajo licencia GNU/GPL (Licencia Pública General) o bien adquiriendo licencias comerciales de MySQL AB en el caso de no desear estar sujeto a los términos de la licencia GPL. Su diseño multi-hilo le permite soportar una gran carga de forma muy eficiente (Gracia, 2005).



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

- Consume muy pocos recursos, tanto de CPU como de memoria.
- Es *Open Source*.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de *API's* en gran cantidad de lenguajes (C, C++, Java, PHP).
- Gestión de usuarios y *passwords*, manteniendo un muy buen nivel de seguridad en los datos.
- Posee mayor rendimiento.
- Mejores utilidades de administración (*backup*, recuperación de errores).
- Mejor integración con PHP.
- No presenta límites en el tamaño de los registros.
- Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos (Curso de MySQL. MySQL con Clase).

PostgresSQL

PostgresSQL es un sistema objeto-relacional que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. Está bajo la licencia BSD (*Berkeley Software Distribution*). La licencia BSD permite el uso del código fuente en software no libre al contrario de la GPL (PostgreSQL, 1996).

Las principales características de este gestor de bases de datos son las siguientes (Eduardo, 2005).

- Soporta el uso de índices, reglas y vistas.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Altamente adaptable a las necesidades del cliente.
- Corre en casi todos los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows, etc.



Algunas Desventajas:

- Es más lento que MySQL.
- Posee menos funcionales en PHP.
- Consume más recursos y carga más el sistema.

1.6.4 Selección del sistema gestor de bases de datos a utilizar.

Para la presente investigación se decidió utilizar MySQL, pues proporciona gran rapidez y seguridad, así como facilidad de uso para los usuarios, además de una excelente integración con el lenguaje PHP, está bajo licencia GNU/GPL y muchas otras características expuestas anteriormente, se decide utilizar MySQL como sistema de gestión de base de datos, para el desarrollo del sistema propuesto.

1.6.5 Servidor Web.

El servidor web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de petición, el servidor web buscará una página web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML (Lenguaje de Marcas de Hipertexto o *HyperText Markup Language* en Inglés) al cliente o navegador que realizó la petición. El servidor web es fundamental en el desarrollo de las aplicaciones del lado del servidor, ya que estas se ejecutan en él. (Ciencia y Técnica Administrativa, 2008)

Apache 2.2.21.

El nombre Apache proviene de “*a patchy server*” que significa “un servidor lleno de remiendos”, esto quiere decir que es un servidor construido con un código preexistente y piezas y parches de código. El servidor web Apache se caracteriza por su robustez y estabilidad, lo que lo convierte en una herramienta confiable (MOHAMMEND, 2003). Entre sus características se destacan:

- Multiplataforma.
- Conforme al protocolo HTTP.



- Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona y con las API⁵ de programación de módulos para el desarrollo de módulos específicos.
- Fiabilidad: Alrededor del 90% de los servidores con más alta disponibilidad funcionan bajo servidores Apache.
- Software Libre: Apache es totalmente gratuito y se distribuye bajo la licencia Apache Software License, que permite la modificación del código. (Gracia, 2005)
- Extensibilidad: se pueden añadir módulos para ampliar las capacidades que posee. (Peña, Rodríguez, & Paredes García, 2008)
- Permite emplear diversos lenguajes en el lado del servidor (PHP, Python y Perl principalmente).
- Permite las conexiones seguras y la utilización de *URL* amigables (MOHAMMEND, 2003).

1.7 Lenguajes de programación utilizados.

En esta sección se especifican los diferentes lenguajes de programación empleados en la realización del producto.

1.7.1 Lenguaje del lado del servidor.


Los lenguajes del lado del servidor son reconocidos, ejecutados e interpretados por el propio servidor web y se envían al cliente en un formato comprensible para él.

PHP 5.3.10.

PHP es un acrónimo recursivo que significa PHP *Hypertext Pre-processor*, (inicialmente se llamó *Personal Home Page Tools*). Fue creado por el programador danés-canadiense Rasmus Lerdorf en 1995 y desarrollado por PHP Group⁶. PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Este lenguaje es publicado bajo la licencia PHP, compatible con la Licencia Pública General de GNU. La mayor parte de su sintaxis ha sido tomada de

Abreviatura de *Application Programming Interface*. Conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación.

⁶ PHP Group: grupo de desarrollo de php.



C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión php (HEURTEL, O, 2011).

Ventajas:

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos base de datos entre ellos MySQL, que es el que se utiliza en el desarrollo del sistema propuesto.
- Capacidad de expandir su potencial utilizando módulos.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Desventajas:

- Se necesita instalar un servidor web, en este caso Apache.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP (Álvarez, Miguel Ángel, 2009).

Partiendo de las ventajas que ofrece PHP se decide utilizarlo para la implementación y desarrollo del sistema. Las desventajas que presenta este lenguaje pueden ser solventadas de manera fácil, al instalar Apache como servidor web y evitando mezclar las sentencias HTML y PHP.

1.7.2 Lenguajes del lado del cliente.

Los lenguajes de programación del lado del cliente se ejecutan en el navegador del usuario. Las páginas del cliente son muy dependientes del sistema donde se están ejecutando y esa es su principal desventaja. Cada navegador tiene sus propias características y lo que puede funcionar en un navegador puede no funcionar en otro. Como ventaja se puede decir que ofrecen respuestas inmediatas a las acciones del usuario y permiten la utilización de recursos de la máquina local (PADILLA, 2008).

HTML 5.

HTML (por sus siglas en inglés *Hypertext Markup Lenguaje* o Lenguaje de Marcas de Hipertexto). Es un lenguaje sencillo, fácil de entender y utilizar. La versión 5 de HTML no se limita sólo a crear nuevas etiquetas, atributos y eliminar las marcas que están en



desuso o se utilizan inadecuadamente; sino que pretende proporcionar una plataforma para desarrollar aplicaciones web más parecidas a las aplicaciones de escritorio (Álvarez, Miguel Ángel, 2009).

Ahora bien HTML define la estructura y el contenido de las páginas permitiendo combinar textos, imágenes, sonidos, vídeos y enlaces a otras páginas. Así como también su nivel de complejidad es bajo y permite incluir dentro de su código scripts, escritos en otros lenguajes como PHP y JavaScript (fresno, V.D, 2006).


Se puede decir que HTML es el lenguaje más usado mundialmente, para la creación de sitios estructurados (Noguera, Bulmaro, 2011). A continuación se exponen algunas de las ventajas de HTML 5:

- Ofrece la posibilidad de obtener un código más limpio y fácil de depurar, que los códigos de los estándares anteriores.
- Inclusión de las etiquetas video y audio. Soporta de manera eficiente y estable cualquier opción de ejecución de video y audio, sin generar errores o añadir código flash en el sitio web.
- Nueva estructura de etiquetas mejorada. Permite definir por separado el encabezado, la barra de navegación, las secciones de la página web, los textos del sitio, los diálogos y el pie de página de los sitios web.

CSS 3.

Es un lenguaje de hojas de estilos, creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. Es una forma de separar los contenidos y su presentación. La utilización del CSS se hace imprescindible, puesto que permite darle estilo a los elementos de las páginas de sistemas web y lograr la interfaz gráfica de la aplicación requerida. Drupal separa el contenido del sitio de la manera en que estos se muestran, esa característica hace necesario el uso de CSS. Sus siglas en inglés significan *Cascade Style Sheet*, que en español quiere decir Hoja de Estilos en Cascadas (SCHULZ, R. G, 2008).

Frameworks de JavaScript JQuery 1.10.



Consiste en un único fichero JavaScript que contiene las funcionalidades comunes del DOM⁷, eventos, efectos y AJAX⁸, permitiendo simplificar la manera de interactuar con los documentos HTML. Al igual que otras librerías, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código. Es decir, con las funciones propias de esta librería se logran resultados en menos tiempo de desarrollo (THE JQUERY FOUNDATION. jQuery is a new kind of JavaScript Library, 2013).

Conclusiones del Capítulo.

- ✓ El análisis de los principales procesos de la DEC en la UCI, permitió constatar la necesidad de la creación de un sistema para la gestión de la información de los mismos.
- ✓ El estudio de las tecnologías de la informática y las comunicaciones y la observación de las distintas tendencias en el desarrollo de software, permitió identificar al CMS Drupal como plataforma de desarrollo, MySql como sistema gestor de bases de datos, PHP como lenguaje de programación del lado del servidor, y como lenguajes del lado del cliente HTML, CSS3, y JavaScript mediante el Framework JQuery.
- ✓ El análisis realizado a sistemas informáticos que implementan alguna solución para satisfacer la gestión de información en determinados centros e instituciones, posibilitó un mayor conocimiento para el futuro desarrollo de la solución.
- ✓ La comparación, el análisis y definición de las diferentes metodologías tanto ágiles como tradicionales, hizo posible la selección de una metodología ágil para el correcto desarrollo de la investigación.
- ✓ Las tecnologías, herramientas, lenguajes de programación y desarrollo analizados, permitieron tener un mayor conocimiento sobre ellos para su posterior utilización.

⁷ *Document Object Model* (Modelo de Objetos del Documento). Define cómo los objetos de una página Web son representados, qué atributos tienen y cómo se manipulan. DOM es una interfaz para que aplicaciones y scripts puedan acceder a un documento pudiendo modificar su estructura y contenido.

⁸ *AJAX (Asynchronous JavaScript and XML)*: Es una técnica de desarrollo web para crear aplicaciones interactivas.



Capítulo 2: Solución Propuesta.

Introducción.

Para un correcto desarrollo de la solución propuesta y utilizando como metodología de desarrollo SXP, se tuvieron en cuenta cada uno de los pasos para el desarrollo de software que propone dicha metodología. Por lo tanto el objetivo del presente capítulo, se basa en la realización de una descripción de las fases definidas por SXP, haciendo énfasis en la fase de planificación, donde se realizan descripciones de algunos artefactos generados por la misma, de los cuales se registran las historias de usuarios y el plan de iteraciones.

2.1 Propuesta de solución.

La presente investigación se centra en el desarrollo de un sistema que gestione toda la información que se genera en él la dirección de extensión de la UCI. El cliente fue explícito al exponer la palpable necesidad de una aplicación lo más minimalista posible en cuanto a detalles estéticos y visuales, en la que la sencillez y la facilidad de su uso fuesen la premisa fundamental que propiciaran un ambiente fluido con la finalidad de optimizar el trabajo. El sistema cuenta con 6 módulos, un módulo inicial que lleva a cabo toda configuración de roles y permisos para los usuarios y administradores. Los requisitos fundamentales planteados inicialmente por el cliente fueron los correspondientes a la gestión de actividades culturales que a su vez forma parte de módulo 2, los restantes módulos (reservaciones, noticias, patrimonio cultural, el salón de la fama) y otras funcionalidades (comentarios, foros debates, etc.) de la aplicación fueron añadidas a partir de varios encuentros. En los mismos se discutieron, a través de un minucioso análisis, las particularidades de estas nuevas opciones de valor agregado y sus capacidades e importancia en relación con las novedosas y más actuales tendencias tecnológicas paralelas al continuo desarrollo web, lo que puso de manifiesto las ventajas que cada una de ellas podría ofrecer de ser finalmente incorporadas.

El sistema brinda a los usuarios que accedan a este, la posibilidad de hacer consultas personalizadas a través de distintos filtros de búsqueda de la información de un artista aficionado o una actividad, donde se le mostrará la planilla correspondiente con los datos, y si es un usuario registrado también tendrá la posibilidad de dejar su opinión a través de la sección dedicada a los comentarios, permitiendo el intercambio y enriquecimiento entre usuarios y especialistas, con este mismo fin y además para establecer un intercambio



más personal entre ambas partes contará con un sistema de mensajería que permitirá en el caso de las reservaciones de ofertas culturales el estado de las mismas, una vez que el usuario la realice. Los especialistas y administradores del sistema serán los encargados de gestionar (adicionar, editar, eliminar, buscar y listar) toda la información que se genere y publique y dar soporte a todas las funcionalidades que ofrecerá el sistema. En caso de que sea necesario también podrá otorgar ciertos privilegios a determinados usuarios referentes al acceso y forma de utilizar las distintas secciones y servicios que brinde la aplicación. La opción de exportar cada planilla de ejemplar a formato pdf, así como la impresión también formará parte del conjunto de funcionalidades disponibles.

El sistema contará con una sección de noticias donde los usuarios podrán estar al tanto de las últimas informaciones culturales y sociales de la universidad, cuba y el mundo. El sistema también brindará información sobre las distintas obras que forman parte del patrimonio cultural así como su ubicación y descripción de la misma. El salón de la fama es otro de los módulos donde se permitirá la gestión de la información de los artistas aficionados y obras más destacadas del MAA en la universidad.

2.2 Sistemas de control de acceso.

Cuando se desarrolla una solución determinada un elemento fundamental que se debe tener en cuenta es el control de acceso de los usuarios, con el objetivo de determinar si el usuario tiene permitido el acceso a los recursos. Los modelos de control de acceso son:

- Control de acceso discrecional (DAC).
- Control de acceso obligatorio (MAC).
- Control de acceso basado en roles (RBAC).

DAC y MAC por sí solos son inadecuados para cubrir las necesidades de la mayor parte de las organizaciones. El modelo DAC es demasiado débil para controlar el acceso a los recursos de información de forma efectiva, en tanto que el MAC es demasiado rígido. El modelo RBAC por su parte, es un intento de unificar los modelos anteriores, consiguiendo un sistema que impone el control de accesos, pero sin las restricciones rígidas impuestas por las etiquetas de seguridad, para esta investigación se definió el RBAC.

2.2.1 RBAC.

El Sistema de control de acceso basado en roles RBAC básicamente consiste en la creación de roles para los trabajos o funciones que se realizan en la organización, su



arquitectura está ideada para asignar permisos a operaciones más que a objetos concretos, siendo por tanto su principal ventaja el permitir alinear la infraestructura de seguridad de la información con los objetivos del negocio de una forma natural. Este recopila además las capacidades de súper usuario (usuario *root*) en perfiles de derechos, los mismos se asignan a cuentas de usuario especiales denominadas roles. Luego, un usuario puede asumir un rol para realizar un trabajo que requiere algunas de las capacidades de súper usuario.

RBAC actualmente es considerado uno de los modelos más generales, debido a su neutralidad respecto a las políticas de control de acceso y a su flexibilidad. A causa de la existencia de múltiples variantes de modelos e implementaciones del mismo se destacan las siguientes características:

- ✓ Administración de autorizaciones:
 - La asignación de permisos a usuarios tiene dos partes: asociar usuarios a roles y asignar permisos para objetos a roles.
 - Si un usuario cambia de tareas, solo basta con cambiarle el rol.
- ✓ Menor privilegio:
 - Permite implementar la política del menor privilegio posible la cual consiste en que un usuario dispone exactamente de la cantidad de privilegios necesaria para realizar un trabajo.
 - Si una tarea no va a ser ejecutada por un usuario, entonces su rol no tendrá los permisos para hacerla, de esta manera se minimizan riesgos de daños.
- ✓ Separación de responsabilidades
 - Se basa en el principio de que ningún usuario tenga suficientes privilegios para usar el sistema en su propio beneficio.

2.2.2 Roles de usuario.

Una de las premisas fundamentales a tener en cuenta cuando se comienza el desarrollo de cualquier sistema informático, la constituye el delimitar la audiencia a la cual va dirigido el mismo, teniendo en cuenta que esta puede estar dividida a su vez en grupos atendiendo a sus competencias. Dentro de la audiencia antes mencionada se incluyen, como personal relacionado al sistema, al administrador que será el encargado de gestionar toda la información, el usuario general y el usuario autenticado. Los usuarios del portal serán aquellos que obtengan un resultado de la ejecución de uno o varios procesos

del sistema. A continuación se muestra un listado con los roles de usuarios definidos para esta aplicación:

Roles	Descripción
Administrador	Es el web máster del sistema, usuario único, encargado de toda la gestión de roles y permisos pertinentes para todo tipo de usuarios en el sistema.
Especialista superior	Es el encargado de la gestión de información dentro del sistema a diferentes niveles, según su cargo, es decir estos vienen siendo director, jefes de departamento, etc.
Usuario autenticado	Tiene acceso a todo lo publicado y definido por el administrador del sistema para los usuarios. Tendrá privilegios extras en dependencia de los permisos que le sean asignados.
Usuario anónimo	Cualquier persona que acceda a la aplicación. Su acceso a la misma es definido por el administrador funcional, así como su nivel de visualización de la información.

Tabla 3. Roles de usuario del Sistema.

2.3 Fases de SXP:

Fase Planificación-definición: En esta fase se define el alcance general del proyecto siendo el cliente quien precisa lo que necesita mediante la redacción de sencillas historias de usuarios, así como un correcto levantamiento de los requisitos. Esta fase dura aproximadamente un par de semanas y el resultado es una visión general del sistema y un plazo total estimado. (Fernandez Céspedes, y otros, 2008).

En esta fase el cliente y los desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario y asociadas a estas las entregas. Esta fase consiste en una o varias reuniones grupales de planificación y el resultado de esta fase es un plan de entregas. Después de ser descritas e identificadas las historias de usuario y estimado el esfuerzo propuesto para la realización de cada una de ellas, se procede a la planificación de la etapa de implementación del sistema. En este plan se especifica



exactamente cuáles historias de usuario serán implementadas para cada iteración del sistema y las posibles fechas para sus liberaciones. En esta fase intervienen los roles: cliente, analista, diseñador, scrum máster, arquitecto, gerente. (Fernandez Céspedes, y otros, 2008).

Fase Desarrollo: Es la segunda fase que define la metodología SXP. En la primera parte de esta fase se generan documentos como:

- ✓ Plan de las iteraciones.
- ✓ Plan de duración de las iteraciones.
- ✓ Estándares de programación.
- ✓ Casos de prueba de aceptación.
- ✓ Tareas de ingeniería a realizar durante la implementación.

Además se genera el código fuente en la etapa de implementación, y luego los documentos relacionados con las pruebas, como última parte de esta etapa, en esta fase intervienen los roles: cliente, programador, tester (probador), gerente. (Fernandez Céspedes, y otros, 2008).

Fase Entrega: Tercera fase que define la metodología SXP, En esta fase se realiza la entrega del software y su documentación, generándose aquellos documentos que son imprescindibles para el entrenamiento y entendimiento del producto como son:


- ✓ Manual de usuario
- ✓ Manual de identidad
- ✓ Manual de desarrollo.

El rol programador es el encargado de esta fase. (Fernandez Céspedes, y otros, 2008).

Fase Mantenimiento: La fase de Mantenimiento, es la cuarta y última que define la metodología SXP. En esta fase se realizan las actividades relacionadas con el soporte del software y se generan los documentos relacionados con los cambios que puedan ocurrir en el mismo. El rol que interviene en esta metodología es el programador. (Fernandez Céspedes, y otros, 2008).

2.4 Especificación de los requisitos de software.

2.4.1 Requisitos funcionales:



Los requisitos funcionales definen el comportamiento interno del sistema, son las funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica. (PRESSMAN, 2002)

RF1: Autenticar usuario: proporciona el acceso inicial al sistema, solo a aquellos usuarios registrados en el mismo, para lo cual, se necesita un formulario con los campos usuario y contraseña. Para poder acceder al sistema todos los usuarios deben estar autenticados. En el caso de que el usuario intente acceder alguna información no autorizada, el sistema debe denegar el acceso.

RF2: Gestionar usuario: este requisito es uno de los encargados de la seguridad del sistema; tiene como funcionalidad el trabajo con los datos de los usuarios autorizados a la manipulación de la información en el sistema.

- ✓ **Adicionar usuario:** para ingresar un nuevo usuario al sistema se requiere de un formulario que contenga los campos nombre de usuario, correo electrónico, contraseña, estado y el rol a desempeñar dentro del sistema.
- ✓ **Editar usuario:** para modificar un usuario existente, se debe cargar un formulario con los datos registrados del usuario.
- ✓ **Eliminar usuario:** para eliminar un usuario debe existir un formulario que muestre a todos los existentes en el sistema y que pueda ser seleccionado aquel que desea eliminarse.
- ✓ **Listar usuario:** se listarán todos los usuarios en dependencia de su rol.
- ✓ **Buscar usuario:** para buscar un usuario se debe introducir el nombre y buscar la correspondencia en la lista de usuarios.

RF3: Gestionar rol: tiene como funcionalidad la creación de nuevos roles, actualización de los ya creados, así como la eliminación de roles que sean desechados por parte del Administrador del Sistema.

- ✓ **Adicionar rol:** para crear un nuevo rol en el sistema se requiere de un formulario que contenga el campo nombre del rol.
- ✓ **Editar rol:** para actualizar un rol existente se debe cargar un formulario para seleccionar el rol a modificar, así como los permisos de acceso a las diferentes funcionalidades de los módulos.



- ✓ **Eliminar rol:** para eliminar un rol, debe existir un formulario que muestre a todos los existentes en el sistema y que pueda ser seleccionado aquel que desea eliminarse. Si hay usuarios asignados a ese rol no se podrá eliminar hasta que no sean eliminados esos usuarios o movidos a otros roles.
- ✓ **Listar rol:** Se listarán todos los roles del área de trabajo en que se encuentre.
- ✓ **Buscar rol:** para buscar un rol se debe introducir el nombre y buscar la correspondencia en la lista de roles.

RF4: Asociar un rol a un usuario: permite la asociación de cada uno de los roles con cada uno de los usuarios, donde un usuario puede tener muchos roles, así como un rol puede estar asignado a varios usuarios.

RF5: Gestionar ofertas culturales: tiene como funcionalidad e trabajo con los datos de las reservaciones.

- ✓ **Adicionar oferta:** para adicionar una reservación al sistema se requiere de un formulario con los campos fecha de la oferta, título, descripción, precio, vigencia (periodo de reservación), fecha de expiración, disponibilidad (cantidad restante) y si incluye transporte o no.
- ✓ **Editar oferta:** para modificar una oferta existente, se debe cargar un formulario con los datos registrados de la misma.
- ✓ **Eliminar oferta:** para eliminar una oferta debe existir un formulario que muestre todas las ofertas existentes en el sistema y que pueda ser seleccionada aquella que desee eliminarse.
- ✓ **Listar oferta:** se listarán todas las ofertas existentes en el sistema.
- ✓ **Buscar oferta:** debe existir un formulario con un campo donde se debe introducir el nombre de la oferta y buscar la correspondencia de la misma en el listado.

RF6: Realizar reservación: en el mismo se realiza la reservación, para la realización de la misma se requiere de un formulario con el listado de las ofertas y de las personas.

RF7: Cancelar Reservación: se carga un formulario con las reservaciones hechas por el usuario, y se selecciona la que se desee cancelar.

RF8: Confirmar Reservación: se carga un formulario con un texto de confirmación, así como se envía un e-mail al usuario con los datos de la misma.



RF9: Gestionar de las actividades culturales: para llevar a cabo la gestión de las actividades culturales se requiere de un formulario con los campos nombre, descripción, fecha, lugar, nivel, objetivo, tipo de público, unidades artísticas, guion, tipología, director artístico y se lleva premiación o no.

- ✓ **Adicionar actividades:** para adicionar una actividad al sistema se requiere de un formulario con los campos fecha de la actividad, nombre, descripción, nivel, lleva premiación, lugar, objetivo/motivo/intención, público, director artístico, guión, premio (opcional), obra(s).
- ✓ **Editar actividades:** para modificar una actividad existente, se debe cargar un formulario con los datos registrados de la misma.
- ✓ **Eliminar actividades:** para eliminar una actividad debe existir un formulario que muestre todas las actividades existentes en el sistema y que pueda ser seleccionada aquella que desee eliminarse.
- ✓ **Listar actividades:** se listarán todas las actividades existentes en el sistema.
- ✓ **Buscar actividades:** debe existir un formulario con un campo donde se debe introducir el nombre de la actividad y buscar la correspondencia de la misma en el listado.

RF10: Gestionar tipos de actividad cultural: para la gestión de los tipos de actividad se necesita de un formulario con un campo, donde se seleccione el tipo de actividad, lo que carga automáticamente una planilla, para la inserción de los datos.

RF11: Gestionar artista aficionado: este requisito tiene como funcionalidad el trabajo con los datos de los artistas aficionados y la manipulación de la información del mismo en el sistema.

- ✓ **Adicionar artista:** para ingresar un nuevo artista al sistema se requiere de un formulario que contenga los campos nombre y apellidos del artista, CI, solapín, facultad, año y género.
- ✓ **Editar artista:** para modificar un artista existente, se debe cargar un formulario con los artistas existentes en el sistema y se selecciona el que se le quieren



- ✓ **Eliminar artista:** para eliminar un artista debe existir un formulario que muestre a todos los artistas existentes en el sistema y que pueda ser seleccionado aquel que desea eliminarse.
- ✓ **Listar artista:** se listarán todos los artistas del sistema.
- ✓ **Buscar artista:** para buscar un artista se debe introducir el nombre y buscar la correspondencia en la lista de de los mismos.

RF12: Gestionar unidades artísticas: en el mismo se lleva a cabo la gestión mediante un formulario que contiene un control de versiones de los datos de las unidades y un listado de los artistas que la componen, también da la posibilidad de exportar la información en archivo pdf.

RF13: Gestionar obras culturales: tiene como funcionalidad el trabajo con los datos de las obras culturales y actualización de las mismas.

- ✓ **Adicionar obras:** para adicionar una obra al sistema se requiere de un formulario con los campos nombre, autor, intérprete y subnivel.
- ✓ **Editar obras:** para modificar una obra existente, se debe cargar un formulario con los datos registrados de la misma.
- ✓ **Eliminar obras:** para eliminar una obra debe existir un formulario que muestre todas las obras existentes en una actividad específica y que pueda ser seleccionada aquella que desee eliminarse.
- ✓ **Listar obras:** se listaran todas las obras existentes en el sistema.
- ✓ **Buscar obras:** debe existir un formulario con un campo donde se debe introducir el nombre de la obra y buscar la correspondencia de la misma en el listado dentro de la actividad.

RF14: Listar artistas aficionados: se listarán todos los artistas que se encuentren en el sistema y brinda la opción de descargar en archivo pdf e imprimir.

RF15: Listar actividades de una unidad artística: se listarán todas las actividades en las que haya participado un artista que se encuentren en el sistema, y brinda la opción de descargar en archivo pdf e imprimir.

RF16: Gestionar premios: permitirá la gestión de los premios en asociación de una unidad artística con una actividad cultural.



- ✓ **Adicionar premio:** para adicionar un premio al sistema se requiere de un formulario con los campos nombre del premio, gestión de categoría.
- ✓ **Editar premio:** para modificar un premio existente, se debe cargar un formulario con los datos registrados del mismo.
- ✓ **Eliminar premio:** para eliminar un premio debe existir un formulario que muestre todos los premios existentes de una unidad artística específica y que pueda ser seleccionado aquel que desee eliminarse.
- ✓ **Listar premio:** se listaran todos los premios existentes en el sistema por unidad artística.
- ✓ **Buscar premio:** debe existir un formulario con un campo donde se debe introducir el nombre del premio y buscar la correspondencia del mismo en el listado dentro de la actividad.

RF17: Publicar artículos de interés: se publicarán todos los artículos en el sistema.

RF18: Archivo de publicaciones: se almacenan todos los artículos del sistema.

RF19: Listar publicaciones por autor: se listarán todos los artículos del sistema de un autor determinado.

RF20: Gestionar comentarios: tiene como funcionalidad el trabajo con los comentarios y opiniones del sitio.

- ✓ **Adicionar comentario:** para adicionar un comentario a un artículo se requiere de un formulario de autenticación con los campos usuario y contraseña.
- ✓ **Editar comentario:** para modificar un comentario existente, se debe cargar un formulario con el contenido del mismo.
- ✓ **Eliminar comentario:** para eliminar un comentario debe existir un formulario que muestre todos los comentarios existentes de un artículo específica y que pueda ser seleccionado aquel que desee eliminarse.
- ✓ **Listar comentario:** se listaran todos los comentarios existentes en el sistema de un artículo determinado.
- ✓ **Buscar comentario:** debe existir un formulario con un campo donde se debe introducir el nombre del premio y buscar la correspondencia del mismo en el listado dentro de la actividad.



RF21: Gestionar obras culturales del patrimonio cultural: tiene como funcionalidad el trabajo con los datos de las obras que forman parte del patrimonio.

- ✓ **Adicionar obra patrimonial:** para adicionar una obra patrimonial al sistema se requiere de un formulario con los campos nombre, descripción, autor, ubicación, foto representativa, archivo de audio o video.
- ✓ **Editar obra patrimonial:** para modificar una obra existente, se debe cargar un formulario con los datos registrados de la misma.
- ✓ **Eliminar obra patrimonial:** para eliminar una obra debe existir un formulario que muestre todas las obras existentes en una actividad específica y que pueda ser seleccionada aquella que desee eliminarse.
- ✓ **Listar obra patrimonial:** se listaran todas las obras existentes en el sistema.
- ✓ **Buscar obra patrimonial:** debe existir un formulario con un campo donde se debe introducir el nombre de la obra y buscar la correspondencia de la misma en el listado dentro de la actividad.

RF22: Gestionar artistas en el salón de la fama: tiene como funcionalidad el trabajo con los datos de los artistas más destacados del movimiento, así como una multimedia con todos los videos, entrevistas y artículos publicados del artista en toda su estancia en la universidad.

- ✓ **Adicionar artista:** para adicionar un artista aficionado al salón de la fama, solo se carga de la base de datos el perfil completo de mismo.
- ✓ **Editar artista:** para modificar los datos de un artista existente, se debe cargar un formulario con todos los datos registrados del mismo.
- ✓ **Eliminar artista:** para eliminar un artista se debe cargar un formulario que muestre a todos los existentes en el salón de la fama, y que pueda ser seleccionado aquel que desea eliminarse.
- ✓ **Listar artistas:** se listaran todos los usuarios del salón de la fama.
- ✓ **Buscar artistas:** para buscar un artista se debe introducir el nombre y buscar la correspondencia en la lista de artistas.

2.4.2 Requisitos no funcionales:



Los requisitos no funcionales son aquellos que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos. Son todos aquellos requisitos que ni describen información a guardar ni funciones a realizar. (PRESSMAN, 2002)

Requerimientos de apariencia o interfaz externa: La apariencia del sistema propuesto debe cumplir con los estándares de diseño establecidos por el centro cultural de la UCI. Se utilizan colores claros, con un contraste agradable a la vista y un tamaño de fuente apropiado para una lectura fácil. La interfaz es amigable, fácil de usar, sencilla, asequible para todo tipo de usuarios, de forma tal que los usuarios sin conocimientos básicos de informática puedan interactuar con ella de forma intuitiva.

Requerimientos de usabilidad: Los usuarios con privilegios de administración deberán tener conocimientos avanzados en los temas referentes a las acciones que se realizan en la gestión de información. El resto de los usuarios se recomienda impartirle un curso básico sobre el manejo e interacción con el sistema que se propone.

Requerimientos de rendimiento: Para poder lograr un rendimiento óptimo del sistema en cuestión, se recomienda que las PC clientes posean 1 giga de RAM y un Navegador que soporte Java script. Además debe permitir trabajar conectados concurrentemente 250 usuarios como mínimo y necesita un servidor de base de datos de (1 o superior) gigas de RAM.

Requerimientos de seguridad: La información manejada por el sistema no será pública, de ahí la necesidad de protegerla de usuarios no autorizados, manipulación inadecuada y estados de inconsistencia. Se implementará de forma tal que cada usuario tenga acceso a la información que esté prevista para su rol. En todo momento la información debe estar disponible para los usuarios con acceso autorizado, y debe preservarse la integridad de los datos.

Restricciones de diseño: Como Plataforma de trabajo contaremos con Drupal 7.26 y Lenguaje de Programación PHP5, o sea utilización de tecnología web y uso de la Arquitectura Modular con sus ventajas y desventajas.

Requisitos de Hardware: El hardware de la estación de trabajo del servidor Web, donde se ejecutará el sistema deberá ser un Procesador Pentium D 2x2 caché. 3.2 GHz y con memoria RAM entre 4 y 8 GB, el servidor de Base de Datos será Procesador Pentium D 2x2 caché. 3.2 GHz con memoria RAM entre 4 y 8 GB y almacenamiento en discos SCSI en espejo y capacidad igual o superior a los 120 GB cada disco. Además deberá contar



con tecnología de respaldo de datos históricos. En la estación de trabajo del cliente se necesita un Procesador Pentium 3 (o superior), con memoria RAM mínima de 256 MB.

Requisitos de software: La PC del cliente debe contar con un Sistema Operativo tanto Windows (win9.x o versión superior) como Linux (cualquiera de sus distribuciones) y el Navegador Web compatible con HTML5 y CSS3, podrá ser *Netscape* 3 (o superior), Internet Explorer 4.2 (o superior), *Mozilla Firefox* (cualquiera de sus distribuciones) y compatibles. En el Servidor Web deberá instalarse Servidor Web Apache 2.2.X o superior y en el Servidor de Base de Datos se necesita MySQL.

Requisitos de Licencia: Licencia bajo código abierto: El sistema debe cumplir con los lineamientos necesarios para la producción de software libre y la comunidad de desarrollo y soporte, debe ser con Licencia Pública General para Bibliotecas de GNU (LGPL por sus siglas en inglés) no puede incluir elementos de Licencia Pública General de GNU (GPL por sus siglas en inglés) dentro del desarrollo.

2.5 Historias de usuario.

La Historias de Usuarios (HU) es uno de los principales artefactos que propone la metodología SXP (Scrum-XP), son tarjetas en las que se recogen de forma esquemática y en un lenguaje claro que es lo que se quiere hacer. Son redactadas desde las perspectivas del cliente con ayuda de los programadores. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. (Fernandez Céspedes, y otros, 2008)

Historia de Usuario	
Número: 2	Nombre Historia de Usuario: Gestionar Usuario
Prioridad en Negocio: muy alta	Referencia: RF2
Usuario: administrador, especialista superior.	
Descripción:	
✓ Adicionar usuario: para ingresar un nuevo usuario al sistema se requiere de un formulario que contenga los campos nombre de usuario, usuario, CI, sexo y el rol a desempeñar dentro del sistema.	
✓ Editar usuario: para modificar un usuario existente, se debe cargar un	




formulario con los datos registrados del usuario.
✓ Eliminar usuario: para eliminar un usuario debe existir un formulario que muestre a todos los existentes en el sistema y que pueda ser seleccionado aquel que desea eliminarse.
✓ Listar usuario: se listarán todos los usuarios en dependencia de su rol.
✓ Buscar usuario: para buscar un usuario se debe introducir el nombre y buscar la correspondencia en la lista de usuarios.

Tabla 2.1. HU. Gestionar Usuario.

Historia de Usuario	
Número: 3	Nombre Historia de Usuario: Gestionar Rol
Prioridad en Negocio: muy alta	Referencia: RF3
Usuario: administrador, especialista superior.	
Descripción: <ul style="list-style-type: none">✓ Adicionar rol: para crear un nuevo rol en el sistema se requiere de un formulario que contenga el campo nombre del rol.✓ Editar rol: para actualizar un rol existente se debe cargar un formulario para seleccionar el rol a modificar, así como los permisos de acceso a las diferentes funcionalidades de los módulos.✓ Eliminar rol: para eliminar un rol, debe existir un formulario que muestre a todos los existentes en el sistema y que pueda ser seleccionado aquel que desea eliminarse. Si hay usuarios asignados a ese rol no se podrá eliminar hasta que no sean eliminados esos usuarios o movidos a otros roles.✓ Listar rol: se listarán todos los roles del área de trabajo en que se encuentre.✓ Buscar rol: para buscar un rol se debe introducir el nombre y buscar la correspondencia en la lista de roles.	

Tabla 2.2. HU. Gestionar Rol.



Nota: Las restantes historias de usuarios están contenidas en el anexo 1 de este documento, así como en la documentación adjunta a este documento.

2.6 Plan de iteraciones.

Una iteración es el acto de repetir un proceso con el objetivo de alcanzar una meta deseada, objetivo o resultados. La planificación de sprint (iteraciones) es una reunión crítica de gran importancia para la metodología. Una planificación mal ejecutada puede arruinar por completo todo el sprint. Este plan tiene como propósito proporcionar al equipo suficiente información facilitando que el equipo trabaje con calma y evitar interrupciones durante unas pocas semanas, además de ofrecerle al cliente suficiente confianza. Una planificación de sprint produce concretamente: una meta de sprint, una lista de miembros y su nivel de dedicación y una lista de historias incluidas en el sprint.

2.6.1 Descripción de las iteraciones.

- **Iteración 1**

Es la iteración la que permite tener la primera versión de prueba, la cual será mostrada al cliente con el objetivo de obtener una retroalimentación para el grupo de trabajo. Será implementada la HU de alta prioridad que constituye la base del módulo brindándole al sistema las funcionalidades básicas.

- **Iteración 2**

En esta iteración será analizada otra historia de usuario de alta prioridad que no fue analizada en la primera iteración. Con la culminación de la misma se tendrá implementada otra de las peticiones del cliente descritas en las HU.

- **Iteración 3**

En esta iteración serán implementadas las funcionalidades de baja prioridad. Estas funciones tienen el propósito de brindar al cliente comodidad en la gestión de otras tareas asociadas a las de alta prioridad como la autenticación y el cambio de contraseña de los usuarios.


2.7 Plan de duración de las iteraciones.

Para una mayor organización del trabajo y como parte del ciclo de vida de un proyecto utilizando la metodología SXP, se crea el plan de duración de cada una de las iteraciones, en este caso se hace para el único equipo de desarrollo con el cual se cuenta. Este plan



tiene como finalidad reflejar la duración de cada iteración, así como el orden en que serán implementadas las HU en cada una de las mismas, lo que ayuda a obtener una idea aproximada del tiempo que durará la confección del sistema en su totalidad.

Iteraciones	Orden de las Historias de usuario a implementar	Duración
Iteración 1	HU2: Gestionar usuario HU3: Gestionar rol HU4: Asociar un rol a un usuario	3 semanas
	HU5: Gestionar de ofertas culturales HU9: Gestionar actividades culturales HU10: Gestionar tipos de actividad cultural	3 semanas
Iteración 2	HU11: Gestionar artista aficionado HU12: Gestionar unidades artísticas HU13: Gestionar obras culturales HU16: Gestionar de premios	3 semanas
	HU20: Gestionar comentarios HU21: Gestionar obras culturales del patrimonio cultural HU22: Gestionar artistas en el salón de la fama	3 semanas
Iteración 3	HU1: Autenticar usuario HU6: Realizar reservación HU7: Cancelar reservación	2 semanas
	HU8: Confirmar reservación HU13: Listar artistas aficionados HU15: Listar actividades de una unidad artística	2 semanas



	HU17: Publicar artículos de interés	2 semanas
	HU18: Archivo de publicaciones	
	HU19: Listar publicaciones por autor	

Tabla.2.3. PI. Plan de duración de las iteraciones.

2.8 Tareas de ingeniería.

Para dar solución a las HU es necesario realizar un grupo de acciones llamadas tareas de ingeniería. Estas son descritas, una vez terminada toda la documentación necesaria para generar el código, en fichas como las historias de usuario, pero en el lenguaje de los programadores y permiten una mejor organización del proceso de implementación del módulo además de posibilitar que sea conocido el grado de complejidad de cada HU teniendo en cuenta la cantidad de tareas asociadas a ella. Se efectúan al final de cada iteración.

Tarea de ingeniería	
Número Tarea: 2	Número Historia de Usuario: HU2
Nombre Tarea: Desarrollar formulario para adicionar usuario.	
Tipo de Tarea: Desarrollo	
Fecha de inicio: 03-02-2014	Fecha de fin: 04-02-2014
Programador Responsable: Adonis Vázquez Campos	
Descripción: Se crea el código que permite crear el formulario para la inserción del usuario en el sistema. El formulario que se crea contiene una serie de datos.	

Tabla 2.4. TI. Desarrollar formulario para adicionar usuario.

Tarea de ingeniería	
Número Tarea: 3	Número Historia de Usuario: HU2
Nombre Tarea: Desarrollar formulario para editar los datos de un usuario.	
Tipo de Tarea: Desarrollo	
Fecha de inicio: 05-02-2014	Fecha de fin: 06-02-2014



Programador Responsable: Adonis Vázquez Campos
Descripción: Es desarrollado el formulario para modificar los datos de los usuarios que se encuentran en el Sistema.

Tabla 2.5. TI. Desarrollar formulario para editar los datos de un usuario.

Tarea de ingeniería	
Número Tarea: 4	Número Historia de Usuario: HU2
Nombre Tarea: Implementar la funcionalidad de eliminar usuarios.	
Tipo de Tarea: Desarrollo	
Fecha de inicio: 07-02-2014	Fecha de fin: 08-02-2014
Programador Responsable: Adonis Vázquez Campos	
Descripción: Se crea el código que permite eliminar un usuario en el sistema.	

Tabla 2.6. TI. Implementar la funcionalidad de eliminar usuarios.

Tarea de ingeniería	
Número Tarea: 5	Número Historia de Usuario: HU2
Nombre Tarea: Implementar la funcionalidad de buscar usuarios.	
Tipo de Tarea: Desarrollo	
Fecha de inicio: 10-02-2014	Fecha de fin: 11-02-2014
Programador Responsable: Adonis Vázquez Campos	
Descripción: Se crea el código que permite buscar un usuario en el sistema.	

Tabla 2.7. TI. Implementar la funcionalidad de buscar usuarios.

Tarea de ingeniería	
Número Tarea: 6	Número Historia de Usuario: HU2
Nombre Tarea: Implementar la funcionalidad de listar usuarios.	
Tipo de Tarea: Desarrollo	
Fecha de inicio: 12-02-2014	Fecha de fin: 13-02-2014



Programador Responsable: Adonis Vázquez Campos
Descripción: Se crea el código que permite listar los usuarios en el sistema.

Tabla 2.8. TI. Implementar la funcionalidad para listar usuarios en el sistema.

Nota: Las tablas restantes de las tareas de la ingeniería se encuentran en el anexo 2 de este documento y en la documentación adjunta al mismo.

2.9 Arquitectura del sistema.

El proceso en el cual se define una solución para los requisitos técnicos y operacionales de un sistema es conocido como diseño de arquitectura. Donde se definen qué componentes conforman el sistema, cómo se relacionan entre ellos, y cómo mediante su interacción llevan a cabo la funcionalidad especificada.

2.9.1 Patrón arquitectónico modular

Drupal es un CMS cuya lógica y funcionalidad está programada en lenguaje PHP, lo cual sigue un modelo de programación estructurada, y a su vez hace uso de un sistema de base de datos relacional es este caso MySQL. El código que constituye el núcleo de Drupal está formado por un conjunto de librerías que permiten gestionar los procesos de arranque del sistema. Estas librerías ofrecen además un conjunto de servicios que permiten integrar las funcionalidades adicionales de los módulos, servicios como conexión y administración de la base de datos, gestión de procesos de mailing (envío), tratamiento de imágenes, internacionalización, soporte para la codificación y un potente entorno de integración de utilidades.

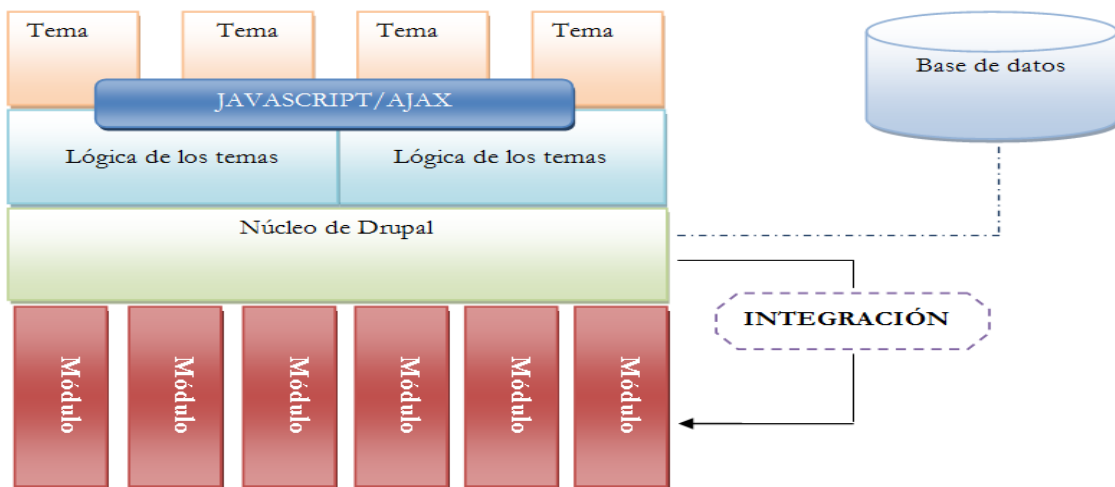




Fig.2.1. Arquitectura de Drupal

Podemos decir que Drupal es un sistema que posee una arquitectura modular que permite ampliar sus funcionalidades a través de unos métodos uniformes de desarrollo e integración de nuevos módulos. Las características que lo distinguen del conjunto de sistemas homólogos son su flexibilidad, escalabilidad, integración con redes sociales, personalización basada en los usuarios, así como la posibilidad de gestión multi-sitio. (Gil, 2012)

2.9.2 Programación orientada a objetos en Drupal.

A pesar de la falta de clases explícitamente declaradas en Drupal, muchos paradigmas orientados a objetos se utilizan en su diseño. Varios son los rasgos necesarios para clasificar como sistema orientado a objetos, algunas de estas definiciones son exhibidas en las características de Drupal como son objeto, abstracción, encapsulación, polimorfismo y herencia. (Colectivo, 2012)

Objetos

Muchas de las construcciones en Drupal se ajustan a la descripción de “objeto” como son los módulos, temas, nodos, y usuarios.

Los nodos son los bloques de construcción básicos de los contenidos de un sitio de Drupal. Los métodos que funcionan en este objeto se definen en *node.module*, usualmente llamado por la función *node_invoke ()*, de manera similar funcionan los objetos de usuario, reuniendo información de cada cuenta en el sitio, información de perfil y sesiones de actividad. En ambos casos, la estructura de datos se define mediante una tabla de base de datos en lugar de una clase. Drupal explota la naturaleza relacional de bases de datos para permitir que otros módulos amplíen los objetos con campos de datos adicionales. (Colectivo, 2012)

En la propuesta de solución la información que se gestiona representan los nodos almacenados que son organizados por los tipos de contenido tales como Actividad Cultural, Artista Aficionado, Unidad Artística. Cada uno de estos tipos de contenido llevan asociado un conjunto de metadatos o atributos que definen los datos que poseen. Ejemplos de nodos son “Festival de la Fac 1”, “La Cumbiamba”. Como representación de los atributos o campos se tiene el caso de Artista Aficionado que contiene la Facultad, el Género y el Año. Las descripciones anteriormente definidas son manipuladas por Drupal haciendo uso de la base de datos en MySQL.



Abstracción

Los hooks definen las operaciones que se pueden realizar en un módulo. Si un módulo implementa un *hook* se entra en un contrato para realizar una tarea en particular cuando el hook se invoca. El código solo necesita saber acerca del módulo o la forma en que el hook se aplica con el fin de obtener un trabajo útil hecho invocando el *hook*. (Colectivo, 2012).

Un ejemplo de los *hook* empleados para el desarrollo del sistema es la implementación del *hook_node_insert* a la hora de crear una reservación se extiende dicha funcionalidad con el objetivo de programar el envío de la notificación de la inserción realizada.

Encapsulación

Como la mayoría de otros sistemas orientados a objetos Drupal no tiene estrictamente una forma de limitar el acceso a un objeto del funcionamiento interno, sino que se basa en la convención para lograr esto. Cada entidad o representación de información en Drupal es tratado con un conjunto de ámbitos de acceso asociados. Esto posibilita definir que roles acceden a cada tipo de entidad y que operación le es permitido realizar sobre dicha información. A través de terceros módulos es posible especificar dichos permisos de acceso para cada instancia de las entidades por separado. (Colectivo, 2012)

En el sistema se gestiona la información por el rol de administrador de contenido que tiene la posibilidad de ingresar, modificar y eliminar todas las entidades del sistema, no siendo así con el resto de los usuarios autenticados que solo puede insertar y modificar las reservaciones de obras culturales.

Polimorfismo

Los nodos son polimórficos en el sentido clásico, si un módulo tiene que mostrar un nodo, por ejemplo, se puede llamar *node_view ()* en ese nodo para obtener una representación HTML. La representación real, sin embargo, dependerá de que tipo de nodo pase a la función, que es directamente análoga a tener la clase de un objeto y determinar su comportamiento cuando se envía un mensaje a ella. Drupal se encarga de las mismas tareas que la introspección de un lenguaje orientado a objetos de la biblioteca en tiempo de ejecución. (Colectivo, 2012)

Un ejemplo representativo de la utilización de este patrón es en el momento de la recuperación de un nodo o entidad específico. El modelo de datos de Drupal no tiene asignado una tabla para cada entidad sino que maneja tablas genéricas por cada uno de



los metadatos asociados. En el momento de ser demandada la información de dicho nodo es utilizada la misma función *node_load* independientemente de su tipo. Esta última función es la encargada de, en dependencia del tipo extender cada uno los metadatos que contiene dicha entidad.

2.10 Patrones de diseño.

Los patrones son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Sus soluciones están basadas en los problemas del diseño que se repiten y que se presentan en situaciones particulares.

Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias. (Mühlrad, 2008)

Patrón Singleton (instancia única)

Patrón creacional que garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia.

Si se piensa en los módulos y temas como objetos y luego siguiendo el patrón Singleton, entonces estos objetos no encapsulan datos, lo que separa a un módulo desde otro es el conjunto de funciones que contiene, por lo que debe ser pensado como una clase con un ejemplo aislado. (Colectivo, 2012)

Un ejemplo de esto es a la hora del módulo rules invocar una función definida por otro módulo por ejemplo la función *node_save* del módulo *node*, utiliza la misma instancia de inyección del módulo que es cargada con el núcleo de Drupal, es decir no tiene que incluir nuevamente la implementación de dicho módulo para hacerlo.

Patrón Decorator (envoltorio)

Patrón estructural encargado de añadir dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender la funcionalidad.

Drupal hace amplio uso de este patrón a través del polimorfismo de nodo de objetos y el uso de *hook_nodeapi ()*, que permite a los módulos extender arbitrariamente el comportamiento de todos los nodos.



Esta característica permite una amplia variedad de comportamientos que se añade a los nodos. Por ejemplo, un nodo comentario tiene solo unas pocas piezas de los datos asociados: título, resumen, cuerpo, usuario, y algunos metadatos. Una necesidad común es que los archivos cargados se adjunten a un nodo, para lo que se podría diseñar un nuevo tipo de nodo con las características de uno de tipo comentario y además con la posibilidad de adjuntar archivos. El módulo *File* de Drupal satisface esta necesidad de forma mucho más modular mediante la concesión de cada nodo que solicite la posibilidad de tener archivos adjuntos como es el caso de las obras de Patrimonio Cultural.

Este comportamiento podría ser imitado por el uso de los decoradores, envolviéndolos alrededor de cada nodo objeto. Más sencillamente, los idiomas que apoyan a las categorías, podrían aumentar la base común de toda clase de objetos nodo para añadir el nuevo comportamiento. La implementación de Drupal es una simple ramificación del sistema de *hooks* y la presencia del *node_invoke_all()*. El caso práctico de utilización de taxonomías para las tipologías o clasificaciones de la información como son las Facultades, los Años, los Géneros que son asociados a los Artistas Aficionados. (Colectivo, 2012)


Patrón Observer (observador)

Patrón de comportamiento encargado de definir un objeto que coordine la comunicación entre objetos de distintas clases, pero que funcionan como un conjunto.

El patrón Observer es generalizado en Drupal. Cuando una modificación es hecha a un vocabulario en el sistema de taxonomía de Drupal, el *hook taxonomy* es llamado en todos los módulos que lo implementan y mediante la aplicación del *hook* se registran como observadores del objeto vocabulario; lo que asegura poder actuar como es apropiado ante cualquier cambio. Ejemplos de taxonomías empleadas en el sistema son el Año, la Facultad, los Premios y los Géneros. (Colectivo, 2012)

Patrón Bridge (puente)

La capa de abstracción de base de datos de Drupal se aplica de una forma similar al patrón estructural Bridge. Los módulos deben ser escritos independientemente del sistema de base de datos que se está utilizando, y la capa de abstracción esta proporcionada para ello. Se pueden definir nuevas capas de dicho sistema de acuerdo a la API definida, añadiendo soporte para otros sistemas sin necesidad de modificar el



código del módulo, desacoplando de esta manera la abstracción de su implementación.
(Colectivo, 2012)

Patrón Chain of Responsibility (cadena de responsabilidad)

Patrón de comportamiento que permite establecer la línea que deben llevar los mensajes para que los objetos realicen la tarea indicada.

En cada solicitud de las páginas, el menú del sistema determina si hay un módulo para gestionar la solicitud, si el usuario tiene acceso a los recursos solicitados y que la función sea llamada para hacer el trabajo. Para ello, el mensaje se pasa a la opción del menú correspondiente a la vía de la solicitud y si el elemento de menú no puede manejar la petición, se pasa a otro eslabón de la cadena. Esto continúa hasta que un módulo se encargue de la petición, un módulo niegue el acceso para el usuario o se haya agotado la cadena. (Colectivo, 2012)

Conclusiones del capítulo

Con la realización de este capítulo se concluye lo siguiente:

- ✓ El estudio de los principales procesos de la DEC en la UCI, posibilitó la definición de la propuesta de solución del sistema.
- ✓ La descripción de los sistemas de control de acceso, brindo el conocimiento adecuado para la utilización de uno de ellos, para mejor seguridad del sistema.
- ✓ La obtención de los requisitos funcionales y no funcionales del sistema permite adquirir una adecuada comprensión de las necesidades existentes y obtener la información necesaria para un posterior desarrollo de la aplicación.
- ✓ El planteamiento de la arquitectura modular posibilitó visualizar el sistema desde diferentes puntos de vistas.
- ✓ La fundamentación de los patrones de diseño a utilizar permite que exista una baja dependencia y una alta reutilización entre las clases. Por consiguiente, una vez precisadas las funcionalidades y la estructura de la solución es posible el comienzo de la implementación del sistema.

Capítulo 3: Implementación y Prueba

Introducción

Este capítulo tiene como objetivo principal mostrar los elementos correspondientes a las etapas de implementación y pruebas del sistema, destacando los casos de prueba como los artefactos fundamentales.

3.1 Estándares de codificación utilizados.

Los estándares de codificación son pautas que se utilizan en la escritura del código fuente, las mismas aseguran de que todos los programadores del proyecto mantengan un vocabulario común, además permite tener un código entendible y organizado. Son muy empleados para asegurar la unificación en el código. Le provee una guía para el encargado del mantenimiento o actualización del sistema, con código claro y bien documentado y además facilita la portabilidad entre plataformas y aplicaciones. (Bermúdez, 2010)

Los patrones de codificación empleados para el desarrollo del sistema son los definidos por el CMS Drupal. A continuación se describen ejemplos de ellos.

3.1.1 indentación

La indentación consiste en insertar espacios en blanco o tabuladores en determinadas líneas de código para facilitar su comprensión. En programación se utiliza para anidar elementos. En Drupal debemos indentar con 2 espacios, nunca con tabuladores. Además, no se debe dejar espacios en blanco al final de cada línea.

En el siguiente ejemplo se muestra un fragmento de código con las indentaciones realizadas, de 2 espacios cada una (->->), y los saltos de línea o Enter () al final de cada línea (sin dejar espacios). (Gil, 2012)

```

<?php
function forum_block_view($delta = '') {
  $query = db_select('forum_index', 'f')
  =>>fields('f')
  =>>addTag('node_access');
  switch ($delta) {
    case 'active':
      $title = t('Active forum topics');
      $query
      =>>orderBy('f.last_comment_timestamp', 'DESC')
      =>>range(0, variable_get('forum_block_num_active', '5'));
      break;
    case 'new':
      $title = t('New forum topics');
      $query
      =>>orderBy('f.created', 'DESC')
      =>>range(0, variable_get('forum_block_num_new', '5'));
      break;
  }
  $block['subject'] = $title;
  // Cache based on the altered query.
  $block['content'] = drupal_render_cache_by_query($query,
    'forum_block_view');
  $block['content']['#access'] = user_access('access content');
  return $block;
}
?>

```

Figura 3.1. Ejemplo de indentación en drupal.

3.1.2 Etiquetas de apertura y cierre de PHP

Cuando estemos escribiendo en PHP, siempre se deben utilizar las etiquetas <?php y ?>, y en ningún caso la versión corta <? y ?>.

En general se omite la etiqueta de cierre de PHP (?>) al final de los archivos .module y .inc. Esta convención evita que se puedan quedar olvidados espacios no deseados al final del archivo (después de la etiqueta de cierre ?>), que serían identificados como salida HTML y podrían provocar un error muy típico, " *Cannot modify header information - headers already sent by...* ".

Por tanto, la etiqueta de cierre final del archivo (?>) es opcional en Drupal. (Gil, 2012).



```
<?php

/**
 * @file
 * Default theme implementation to display a forum which may contain forum
 * containers as well as forum topics.
 *
 * Variables available:
 * - $forums: The forums to display (as processed by forum-list.tpl.php)
 * - $topics: The topics to display (as processed by forum-topic-list.tpl.php)
 * - $forums_defined: A flag to indicate that the forums are configured.
 *
 * @see template_preprocess_forums()
 * @see theme_forums()
 */
?>
<?php if ($forums_defined): ?>
<div id="forum">
  <?php print $forums; ?>
  <?php print $topics; ?>
</div>
<?php endif; ?>
```

Fig 3.2. Etiquetas de apertura y cierre de php.

3.1.3 Operadores

Los operadores binarios, que se utilizan entre dos valores, deben separarse de estos valores, a ambos lados del operador, por un espacio. Por ejemplo, \$numero = 3, en lugar de \$numero=3. Esto se aplica a operadores como +, -, *, /, =, ==, !=, >, <, . (concatenación de cadenas), .+, +=, -=, etc.

Los operadores unarios como ++, -- no deben tener separación. Por ejemplo, \$numero++. (Gil, 2012)


3.1.4 Uso de comillas

Se pueden usar tanto las comillas simples ('cadena') como las comillas dobles ("cadena") para delimitar las cadenas de caracteres.

Las comillas dobles son necesarias si se desean incluir variables dentro de las cadenas de texto. Por ejemplo, "<h1>\$title</h1>". También se recomienda el uso de comillas dobles cuando el texto puede incluir alguna comilla simple. (Gil, 2012)

3.1.5 Uso de punto y coma (;) en código PHP

Aunque PHP permite escribir líneas de código individuales sin el terminador de línea (;), como por ejemplo <?php print \$title ?>. En Drupal es siempre obligatorio: <?php print \$title; ?>. (Gil, 2012)



- Correcto: `<?php print $title; ?>`

- Incorrecto: `<?php print $title ?>`

3.2 Diseño del sistema.

Con el objetivo de la comprensión de este trabajo se hace una breve explicación del funcionamiento de Drupal como plataforma de publicación.

El CMS Drupal contiene un tipo de contenido genérico llamado *Node* que puede ser extendido por cualquier desarrollador. Este tipo de contenido tiene las propiedades básicas para distintas publicaciones como son título, autor, fecha de creación y contenido; además, proporciona los mecanismos para la creación, edición y publicación de este tipo de contenido. Los desarrolladores que deseen una publicación personalizada, solo deben extender el *Node* y de esta manera, aprovechar sus propiedades. Drupal provee al desarrollador de un potente sistema de seguridad basado en roles, el mismo CMS se encarga de la creación de usuarios y roles, así como del control de accesos a los diferentes módulos según los permisos definidos por el administrador. El desarrollador se limita a exportar en su módulo los tipos de acceso que desea definir, el resto lo maneja Drupal, es decir, cuando un usuario trata de acceder a un módulo, la plataforma chequea que el usuario autenticado tenga acceso al módulo en cuestión (Gil, 2012).

3.3 Módulos de Drupal.

Los módulos son el principal mecanismo para extender las funcionalidades de Drupal. Implementan una interfaz bien definida que permite la interacción con el Núcleo. Un módulo es la unión de varias funciones que se juntan en Drupal y ayudan a ofrecerle mayor funcionalidad a la web. Drupal llama *hooks* a las funciones que forman esta interfaz. Los *hooks* los podemos agrupar en 3 categorías: -Autenticación: proveen mecanismos adicionales de autenticación de usuarios. -Núcleo (*Core*): necesario para responder e interactuar con el corazón de Drupal. -*Node*: provee un nuevo tipo de nodo al sistema. Un módulo para Drupal consta de uno o más ficheros, el fichero principal con extensión *.module*, *.info*, *.install*. Básicamente existen dos tipos de módulos: los módulos de contenido, que son los que definen un nuevo tipo de contenido personalizado y la funcionalidad para su creación, edición y publicación y los módulos funcionales, estos tienen disímiles propósitos dependiendo del objetivo con el que se desarrollan. La tarea de estas funciones es actuar como enganche, al ser llamadas por Drupal a la hora de



construir una página web y gestionar el contenido. El sistema de bloques de Drupal consiste en una serie de bloques (ya sean definidos por un módulo o en la misma interfaz de Drupal) que se activan o desactivan para ser mostrados en las áreas de menú de la plataforma. Se utilizará el término “paquete” para referirse a módulos del sistema de gestión de contenidos en cuestión. (Gil, 2012)

Fueron agregados a la plataforma los módulos: validación, autenticación uci, así como la definición de los casos de pruebas unitarias en el modulo test.

3.4 Modelo de implementación.

El modelo de implementación describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables, entre otros. El modelo de implementación describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y como dependen los componentes unos de otros. El modelo de implementación define una jerarquía. (Jacobson, y otros, 2000)

3.5 Fase de Implementación.

Durante el transcurso de las iteraciones, se realiza la implementación de las historias de usuario seleccionadas para ser realizadas en cada una de ellas. Al principio de estas se lleva a cabo una revisión del plan de iteraciones y se modifica en caso de ser necesario. Como parte de este plan, se descomponen las HU en tareas de desarrollo, asignando a un grupo de desarrollo (o una persona), la responsabilidad de su implementación. Estas tareas son para el uso estricto de los programadores, pueden ser escritas en lenguaje técnico y no necesariamente entendible por el cliente. (H. Canós, y otros, 2008)

3.6 Fase de pruebas.

Unos de los pasos más importantes de la metodología SXP es el proceso de pruebas, el cual anima a los desarrolladores a probar constantemente la aplicación tanto como sea posible. Mediante esta filosofía se reduce el número de errores no detectados, así como el tiempo entre la introducción de este en el sistema y su detección. Todo esto contribuye a elevar la calidad de los productos desarrollados y a la seguridad de los programadores, a la hora de introducir cambios o modificaciones. La metodología SXP divide las pruebas en dos grupos: pruebas unitarias; desarrolladas por los programadores, encargadas de



verificar el código de forma automática y las pruebas de aceptación; destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente. (Nazcasoft., 2010) (Blog, 2008)

3.6.1 Pruebas unitarias en la metodología SXP.

Las pruebas unitarias (*unit testing*) son aquellas pruebas que se les hace a los métodos o procedimientos, los cuales deben ser unidades estructurales del programa encargados de una tarea específica. Estas pruebas aisladas proporcionan cinco ventajas básicas, fomentan el cambio, simplifican la integración, documentan el código, separan la interfaz del código y hacen que los errores estén más acotados y sean fáciles de localizar (Nazcasoft, 2010).

En drupal las pruebas unitarias se realizan de manera particular, las pruebas se orientan a las funcionalidades del sistema, en vez de al código del mismo, lo que significa que se escriben de manera tal que se realicen a nivel de interfaz, probando el correcto funcionamiento de todas y cada unas de las funcionalidades desarrolladas. Para la realización de las mismas drupal provee un modulo denominado simple test, el cual permite la creación de los casos de prueba para cada una de las funcionalidades requeridas, de la misma manera que brinda un entorno de trabajo para la ejecución y validación de los resultados obtenidos (Drupal.org, 2013).

3.6.2 Pruebas de aceptación en la metodología SXP.

Las pruebas de aceptación son pruebas de caja negra que se crean a partir de las historias de usuario. Durante las iteraciones las HU seleccionadas serán traducidas a pruebas de aceptación. En ellas se especifican, desde la perspectiva del cliente, los escenarios para probar que una HU ha sido implementada correctamente, además es responsable de verificar el pasaje de las pruebas de aceptación y priorizar la corrección de las pruebas fallidas. Una HU puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo final de estas es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable. Una HU no se considera completa, hasta que no ha pasado por las pruebas de aceptación. (Nazcasoft., 2010) (Blog, 2008)

Caso de Prueba de Aceptación



Código: HU1_P1	Historia de Usuario: 1
Nombre: autenticar usuario	
Nombre de la persona que realiza la prueba: Adonis Vázquez Campos	
Descripción: prueba para la funcionalidad Autenticar usuario.	
Condiciones de Ejecución: El usuario debe existir en la BD.	
Entrada / Pasos de Ejecución Se introduce el usuario y contraseña y se ejecuta la acción Aceptar.	
Resultado Esperado: El usuario accede a la información según los permisos del rol que tiene asignado.	
Evaluación de la Prueba: prueba satisfactoria.	

Tabla 3.3. Prueba 1 al Módulo Configuración.

Caso de Prueba de Aceptación	
Código: HU2_P2	Historia de Usuario: 2
Nombre: gestionar usuario	
Nombre de la persona que realiza la prueba: Adonis Vázquez Campos	
Descripción: prueba para la funcionalidad de adicionar un nuevo usuario en la aplicación.	
Condiciones de Ejecución: cualquier usuario anónimo puede acceder al registro de sus datos desde cualquier parte del sitio a través de un botón ubicado en la parte superior derecha del banner.	
Entrada / Pasos de Ejecución: se intenta añadir los datos requeridos por el formulario de adición de usuario.	
Resultado Esperado: se debe adicionar el usuario de manera correcta en la base de	



datos para su posterior autenticación mostrando un mensaje de aviso e indicando que el proceso fue satisfactorio. En caso que el usuario ignore algunos campos de carácter obligatorio o introduzca datos incorrectos se debe mostrar un mensaje informándole sobre los errores en los que pudo incurrir y además comunicándole que no se pudo completar el proceso de adición.

Evaluación de la Prueba: prueba satisfactoria.

Tabla 3.4. Prueba 2 al Módulo Gestionar Usuarios.

3.6.3 Resumen de pruebas.

A lo largo del desarrollo de todo el sistema se realizaron los casos de prueba, con el objetivo de comprobar la eficiencia del código escrito y el cumplimiento de los requisitos funcionales planteados por el cliente. Por cada historia de usuario se realizó un caso de prueba, salvo algunas excepciones a las que por su complejidad y tamaño se les realizó más de uno. Para un total de 21 historias de usuario se realizaron 57 casos de prueba, de manera que fueron comprobados todos los posibles escenarios, confirmando que los resultados arrojados de las estimaciones muestran un adecuado nivel de similitud ante los resultados reales.

3.7 Impacto de la solución propuesta

Desde el punto de vista el sistema está pensado con el principal objetivo de publicarlo a disposición de los internautas; ofreciendo a todo interesado en la historia e información referente a las tradiciones culturales de Cuba y la UCI, dentro y fuera del país, un total acceso mediante consultas a la ficha completa de cada aficionado y cada obra cultural. Además, constituye un medio para promover la comunicación educativa entre usuarios y especialistas en el tema con fines culturales, investigativos y científicos.

Otra arista y no menos importante es el impacto ambiental, el cual se manifiesta desde dos alcances fundamentales:

- **A corto plazo.**

Se ponen de manifiesto detalles técnicos, que influirán de manera directa en cuanto el cliente acceda al sistema, como:



- ✓ Detalles estéticos y visuales sencillos permitiendo la facilidad de uso logrando así la fidelidad del usuario.
- ✓ La utilización de estándares de diseño como: párrafos cortos, fuente legible y correcto uso de los colores y sus degradados, permitiendo que el usuario revise toda la información sin que se convierta en un proceso de lectura tedioso, y que los detalles visuales no lo afecten a nivel subconsciente.

A largo plazo.

Se hace referencia al PRONAC-EU específicamente a una de sus principales funciones la cual plantea, **“diseñar las acciones extensionistas que contribuyan al cumplimiento de los programas de Educación Ambiental; Prevención en la lucha contra las drogas y otras adicciones; Lengua Materna y al aprendizaje de la Historia y Cultura Nacional”**, es por ellos que la presente investigación que tiene como objetivo el desarrollo de un sistema para la gestión de información de los procesos de la DEC en la UCI, permitirá a los usuarios conocer sobre las acciones que se realizan dentro de los proyectos socioculturales, con vistas a la educación sexual, al uso correcto de la Lengua Materna y al incremento de la calidad de vida de cada uno de los ciudadanos(Gonzales Gil, 2004).

El sistema cuenta con el aval de la Dirección de Extensión de la UCI.

Conclusiones del capítulo.

Una vez finalizado el capítulo se concluye que:

- ✓ La descripción de la propuesta de solución, nos brinda la posibilidad de conocer como el sistema cumple con las exigencias del cliente en cuanto a: usabilidad, diseño y optimización del flujo de datos.
- ✓ Los estándares de codificación los cuales permitieron la organización del código, así como que sea entendible, logrando un vocabulario común entre los programadores.
- ✓ Se realizó el diseño del sistema lo que permitió conocer del funcionamiento de Drupal como plataforma de publicación y como los elementos del modelo del diseño, así como clases, entre otros, fueron implementados.



- ✓ La fase de implementación permitió el desarrollo de las historias de usuario, además de la revisión del plan de iteraciones (la asignación a una persona de la implementación).
- ✓ Las pruebas de aceptación realizadas hasta el momento, han permitido verificar y validar los requisitos identificados, arrojando que la solución propuesta proporciona un avance en la ejecución del proceso de gestión de información de la dirección de extensión de la UCI.
- ✓ La publicación del sistema para los internautas el mismo se convierte en un medio para promover la comunicación educativa entre usuarios y especialistas en el tema con fines culturales, investigativos y científicos.

Conclusiones Generales.

Una vez terminado el presente trabajo de diploma se puede concluir que se desarrollaron todas las tareas a fin de cumplir los objetivos propuestos, resaltando que:

- ✓ Tras el estudio de los principales procesos de la DEC de la UCI, se elaboró un marco teórico y el estudio del estado del arte, partiendo de la necesidad de desarrollar un sistema para la gestión de la información de estos procesos.
- ✓ La realización del análisis y diseño permitió obtener los requisitos funcionales y modelos necesarios para la implementación de las funcionalidades del Sistema Web de la Dirección de Extensión Cultural de la UCI.
- ✓ La implementación del diseño propuesto permite que la Dirección de Extensión Cultural de la UCI cuente con un sistema para el intercambio, gestión y publicación de la información.
- ✓ Las pruebas de software permitieron detectar y corregir los errores no identificados durante la implementación, posibilitando cumplir con las especificaciones requeridas y la validación del sistema implementado, las mismas arrojaron resultados satisfactorios demostrando la calidad del sistema desarrollado.



Recomendaciones.

Una vez terminado este trabajo se recomienda:

- ✓ Para el desarrollo de futuros sistemas para el intercambio, gestión y publicación de información, sea utilizado el presente trabajo como referencia.
- ✓ Se recomienda la puesta en explotación del sistema en el área de Extensión Cultural de la UCI.

Referencias bibliográficas

- Domínguez Fernández, G., & Llorente, M. (2009).** La educación social y la web 2.0 Nuevos espacios de innovación e interacción social en la Educación Superior. *Revista de Medios y Educación.*
- Álvarez, Miguel Ángel. (2009, octubre 14).** *desarrolloWeb.com.* From <http://www.desarrolloweb.com/articulos/que-es-html5.html>
- Gonzales Gil, Ramón. (2004).** *PRONAC-EU (Programa Nacional de Extensión Universitaria).* La habana.
- Ayllapan, W. U. (n.d.).** *www.joomlaos.net.* From <http://www.joomlaos.net/caracteristicas-de-joomla.php>
- Castro Ruz, F. (1986).** *Informe Central al III Congreso del PCC.* La habana: Editora Política. Retrieved noviembre 19, 2013
- Eduardo. (2005, mayo 31).** *tiendalinux.com.* Retrieved febrero, 2014 from http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html
- Ferrer, D. (2007).** Retrieved enero, 2014 from <http://www.maestrosdelweb.com/editorial/que-es-joomla/>
- Figuroa, R., Solís, C., & Cabrera, A. (2008).** *Metodologías Tradicionales Vs. Metodologías Ágiles.* Retrieved enero, 2014 from <http://adonisnet.files.wordpress.com/2008/06/articulo-metodologia-de-sw-formato>
- fresno, V.D. (2006).** *Representación Autocontenida de Documentos HTML: una propuesta basada en Combinaciones Heurísticas de Criterios (Tesis Doctoral).* Móstoles, España. Retrieved febrero, 2014
- García Gómez, Juan Carlos . (n.d.).** *Los portales de internet.*
- Gracia, J. (2005, junio 19).** *www.webestilo.com.* Retrieved febrero, 2014 from <http://www.webestilo.com/mysql/>
- HEURTEL, O. (2011).** *PHP 5.3 Desarrollar un sitio web Dinámico e interactivo.* Barcelona, España: ENI. Retrieved febrero, 2014
- informaticamilenium.com.mx. (1998_2006).** Retrieved febrero, 2014 from Informática Milenium: <http://www.informaticamilenium.com.mx/paginas/espanol/sitioweb.htm>



jquery.com. (2013). Retrieved febrero, 2014 from <http://jquery.com/>

Merelo Guervós, J. J. (2005, marzo 14). *Sitio Web Personal*. Retrieved enero, 2014 from Introducción a los sistemas de gestión de contenidos: <http://geneura.ugr.es/~jmerelo/tutoriales/cms/>.

MOHAMMEND, K. J. (2003). *La biblia del servidor Apache*. Madrid: Cambridge OReilly: Anaya Multimedia. Retrieved febrero, 2014

Noguera, Bulmaro. (2011, julio 4). *culturacion.com*. Retrieved febrero, 2014 from <http://culturacion.com/2011/07/ventajas-de-utilizar-html5/>

PADILLA, R. M. (2008, febrero 28). Retrieved febrero, 2014 from <http://zenkius.blogspot.com/2008/02/tecnologias-del-lado-del->

Peña, H., Rodriguez, Carlos, & Dr. Paredes García, Fernando. (n.d.). Retrieved enero, 2014 from <http://www.drupal.org.es/caracteristicas>

postgresql.org. (1996). Retrieved febrero, 2014 from PostgreSQL: <http://www.postgresql.org/>

Reyero, J. A. (2005, abril 11). *www.drupal.org.es*. Retrieved febrero, 2014 from <http://www.drupal.org.es/caracteristicas>

RODRÍGUEZ, F. (2012). *Experto en Drupal 7*. España: Forcontu S.L.

SCHULZ, R. G. (2008). *Diseño Web con CSS*. Barcelona, España. Retrieved febrero, 2014

SCOTT, K. (2009). *El Proceso Unificado Explicado*. Brasil: Pearson Education. Retrieved enero, 2014

Drupal.org. (2013). <https://www.drupal.org/node/394888>.



Bibliografía

- ✓ **Abraham.** MySQL vs PostgreSQL. ¿Cuándo emplear cada una de ellas? [Online] javaHispano, 2007. [Cited: diciembre 16, 2013.] http://www.javahispano.org/contenidos/es/mysql_vs_postgresql_cuando_emplear_cada_una_de_ellas_11.
- ✓ **Alfonso Lorenzo, Ana Belén Martínez Piñeiro, Esther Martínez Piñeiro.** Revista Electrónica de Investigación y Evaluación Educativa. *ALTIA Consultores*. [Online] Universidade de Santiago de Compostela (España), diciembre 21, 2004. [Cited: enero 10, 2014.] http://www.uv.es/RELIEVE/v10n2/RELIEVEv10n2_6.htm.
- ✓ **Gil, gonzales Ramón.** Programa Nacional de Extensión Universitaria. La habana: s.n., 2004.
- ✓ **Beck, Kent.** Extreme Programming Explained: Embrace Change. Addison Wesley. [Online] septiembre 28, 2009. <http://www.extremeprogramming.org>.
- ✓ **C. Castilla Plaza, Dr. Santodomingo Garachana.** Implicaciones de las tecnologías de la información en la gestión del sistema empresa. *Departamento de Economía Financiera y Contabilidad I*. Universidad Complutense de Madrid : s.n., 1999.
- ✓ **Craig, Larman.** UML y patrones. s.l. : Prentice Hall, 2004. Vol. s.l. ISBN 978-8420534381.
- ✓ **Dolores Lanckenau, Cleopatra Garza.** Sistema Tecnológico de Monterrey. *Dirección de Desarrollo Académico*. [Online] 2012. [Cited: marzo 21, 2012.] <http://www.mty.itesm.mx/rectoria/dda/usols/creacion.htm>.
- ✓ **Fersko-Weis, H.** The case for improved software design. Lotus. 1989. Vol. 5, 8. ISSN 8756-7334.
- ✓ **Guevara, Humberto Rivero.** Trabajo de Diploma: Análisis, diseño e implementación del módulo Aprehensión del SIIPOL. Ciudad de la Habana : s.n., 2008. S.N.
- ✓ **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El Proceso Unificado de Desarrollo de Software*. Madrid : Addison Wesley, 2000.
- ✓ **J. Rumbaugh, I. Jacobson.** El Lenguaje Unificado de Modelado. Manual de Referencia. s.l. : WESLEY, 2000. ISBN 84-7829-037-08.
- ✓ **Jiménez, Arianna Rodríguez.** *Proceso de evaluación de la documentación para el servicio de Pruebas Funcionales a nivel de Sistema que ofrece el Grupo de Calidad del centro FORTES*. Habana : s.n., 2011. S.N.



- ✓ **Jorge Amado Soria Ramírez, Enrique José Altuna Castillo.** Trabajo de Diploma: Análisis, Diseño e Implementación de un Jurado Online. Ciudad de la Habana : s.n., 2008. S.N.
- ✓ **José H. Canós, Patricio Letelier, María Carmen Penadés.** Metodologías Ágiles en el Desarrollo de Software. Valencia : s.n. S.N.
- ✓ **José H. Canós, Patricio Letelier, M^a Carmen Penadés.** Todo Agil. Valencia : DSIC -Universidad Politécnica de Valencia.
- ✓ **Keating, Brian.** DatabaseSpecialists. [Online] [Cited: marzo 10, 2013.] http://www.dbspecialists.com/files/presentations/mm_replication.html.
- ✓ **Kniberg, Henrik.** Scrum y XP desde las trincheras. s.l. : C4Media, 2007. Vol. s.l. ISBN: 978-1-4303-2264-1.
- ✓ **Mestras, J. Pavón.** Patrones de Diseño Orientado a Objetos. [Online] Universidad Complutense de Madrid, 2004. <http://www.fdi.ucm.es/profesor/jpavon/poo/2.14PDOO.pdf>.
- ✓ **Mühlrad, Daniel.** Patrones de diseño. 2008.
- ✓ **Nazcasoft.** Pruebas Unitarias. [Online] CalidadSoftware. [Cited: mayo, 2014.] Disponible en: http://www.calidadyssoftware.com/testing/pruebas_unitarias1.php.
- ✓ **Patricio Letelier, M^a Carmen Penadés.** Metodologías ágiles para el desarrollo de software: Extreme Programming (XP). Valencia : Universidad Politécnica de Valencia. S:N.
- ✓ **Pérez, Javier Eguiluz.** CCS 2.1. 2008.
- ✓ **Pino, José Luis López.** lopezpino. *Servidores web más usados*. [Online] julio 30, 2010. [Cited: enero 22, 2014.] <http://lopezpino.es/2010/07/30/servidores-web-mas-usados>.
- ✓ **Pressman, Roger.** *Ingeniería del Software, un enfoque práctico 5ta Edición*. s.l. : McGraw-hill, 2001. ISBN: 8448132149.
- ✓ **R. Figueroa, C. Solís, A. Cabrera.** Metodologías Tradicionales Vs. Metodologías Ágiles. [Online] 2008. [Cited: enero 24, 2014.] <http://adonisnet.files.wordpress.com/2008/06/articulo-metodologia-de-sw-formato>.
- ✓ **Reynoso, Billy Carlos.** Metodos Agiles en Desarrollo de Software, Introducción a la Arquitectura de Software. [Online] [Cited: enero 25, 2014.] <http://www.microsoft.com/spanish/msdn/arquitectura>.
- ✓ **Roberth Figueroa, Camilo Solis, Armando Cabrera.** Metodologías tradicionales vs metodologías ágiles. *Universidad Técnica Particular de Loja*. s.l. : Escuela de Ciencias en Computación, 2007. S.N.



- ✓ **Roelvis Coello Abijana, Kárel Adrián Silveira García.** Multimedia para Graduados de la Universidad de Ciencias Informáticas. Ciudad de la Habana : s.n., 2007. S.N.
- ✓ **Serrano, Jorge.** Explicando Scrum a mi abuela. *MVP. Visual Developer.* [Online] 2007. [Cited: febrero 5, 2014.] <http://geeks.ms/blogs/jorge/archive/2007/05/09/explicando-scrum-a-mi-abuela.aspx>. S.N.
- ✓ **Sommerville, Ian.** Ingeniería del Software. s.l. : Pearson Education, 2005. Vol. S.I.
- ✓ **Suárez, Danais Rojas.** *Desarrollo del Sitio Web para los profesores guías de la Facultad 8.* Ciudad de La Habana : s.n., 2010. S.N.
- ✓ **Torrens, R.** Arquitectura de la información. [Online] 1999. [Cited: enero 15, 2014.] <http://oai.saber.ula.ve/db/ssaber/Edocs/postgrados/computacion/Arquitecturadelainformacion.pdf>.
- ✓ **Valdés, Damián Pérez.** Los diferentes lenguajes de programación para la web. [Online] [Cited: febrero 12, 2014.] <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web>.
- ✓ **Varios.** Somos Libres. [Online] SurNoticias. Perú, enero 6, 2011. [Cited: mayo 6, 2014.] <http://www.somoslibres.org/>.
- ✓ **Varios.** GIMP. *GNU Image Manipulation Program.* [Online] GIMP, 2001. [Cited: mayo 14, 2014.] <http://www.gimp.org/>. S.N.
- ✓ **Vázquez, J. A. Gallego.** Desarrollo Web con PHP y MySQL. Madrid : Grupo Anaya S.A., 2003. ISBN 84-415-1525-5.
- ✓ **Vegas, Jesús.** Departamento de informática. [Online] Universidad de Valladolid. [Cited: diciembre 20, 2013.] <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html>.
- ✓ **conclase.net. (n.d.).** From <http://mysql.conclase.net/curso/>

Anexos

Anexo 1. Historias de usuario.

Historia de Usuario	
Número: 4	Nombre Historia de Usuario: Asociar un rol a un usuario.
Prioridad en Negocio: muy alta	Referencia: RF4
Usuario: administrador, especialista superior.	
Descripción: permite la asociación de cada uno de los roles con cada uno de los usuarios, donde un usuario puede tener muchos roles, así como un rol puede estar asignado a varios usuarios.	

Tabla 4. HU. Asociar un rol a un usuario.

Historia de Usuario	
Número: 5	Nombre Historia de Usuario: Gestionar de ofertas culturales.
Prioridad en Negocio: muy alta	Referencia: RF5
Usuario: administrador, especialista superior.	
Descripción: <ul style="list-style-type: none">✓ Adicionar oferta: para adicionar una reservación al sistema se requiere de un formulario con los campos fecha de la oferta, título, descripción, precio, vigencia (periodo de reservación), fecha de expiración, disponibilidad (cantidad restante) y si incluye transporte o no.✓ Editar oferta: para modificar una oferta existente, se debe cargar un formulario con los datos registrados de la misma.✓ Eliminar oferta: para eliminar una oferta debe existir un formulario que muestre todas las ofertas en el sistema y que pueda ser seleccionada aquella que desee eliminarse.✓ Listar oferta: se listarán todas las ofertas existentes en el sistema.✓ Buscar oferta: debe existir un formulario con un campo donde se debe introducir el nombre de la oferta y buscar la correspondencia de la misma en el listado.	

Tabla 5. HU. Gestión de ofertas culturales.

Historia de Usuario	
Número: 6	Nombre Historia de Usuario: Realizar reservación.
Prioridad en negocio: alta	Referencia: RF6
Usuario: todos los usuarios que accedan al sistema.	
Descripción: en el mismo se realiza la reservación, para la realización de la misma se requiere de un formulario con el listado de las ofertas y de las personas.	

Tabla 6. HU. Realizar reservación.

Historia de Usuario	
Número: 7	Nombre Historia de Usuario: Cancelar reservación.
Prioridad en negocio: alta	Referencia: RF7
Usuario: todos los usuarios que accedan al sistema.	
Descripción: se carga un formulario con las reservaciones hechas por el usuario, y se selecciona la que se desee cancelar.	


Tabla 7. HU. Cancelar reservación.

Historia de Usuario	
Número: 8	Nombre Historia de Usuario: Confirmar reservación.
Prioridad en negocio: alta	Referencia: RF8
Usuario: todos los usuarios que accedan al sistema.	
Descripción: se carga un formulario con un texto de confirmación, así como se envía un e-mail al usuario con los datos de la misma.	

Tabla 8. HU. Confirmar reservación.

Anexo 2. Tareas de ingeniería.

Tarea de ingeniería



Número Tarea: 7	Número Historia de Usuario: HU3
Nombre Tarea: Desarrollar el formulario para adicionar un nuevo rol al sistema.	
Tipo de Tarea: Desarrollo	
Fecha de inicio: 14-02-2014	Fecha de fin: 15-02-2014
Programador Responsable: Adonis Vázquez Campos	
Descripción: Se crea el código que permite crear el formulario para la inserción del rol en el sistema. El formulario que se crea contiene una serie de datos.	

Tabla 7. Desarrollar el formulario para adicionar un nuevo rol.

Tarea de ingeniería	
Número Tarea: 8	Número Historia de Usuario: HU3
Nombre Tarea: Implementar la funcionalidad para editar los datos de un rol.	
Tipo de Tarea: Desarrollo	
Fecha de inicio: 16-02-2014	Fecha de fin: 17-02-2014
Programador Responsable: Adonis Vázquez Campos	
Descripción: Se crea el código que permite actualizar los datos de un del rol en el sistema.	

Tabla 8. Implementar la funcionalidad para editar los datos de un rol.

Tarea de ingeniería	
Número Tarea: 9	Número Historia de Usuario: HU3
Nombre Tarea: Implementar la funcionalidad para eliminar un rol del sistema.	
Tipo de Tarea: Desarrollo	
Fecha de inicio: 18-02-2014	Fecha de fin: 19-02-2014
Programador Responsable: Adonis Vázquez Campos	
Descripción: Se crea el código que permite eliminar un rol en el sistema.	

Tabla 9. Implementar la funcionalidad para eliminar un rol del sistema.

Tarea de ingeniería	
Número Tarea: 10	Número Historia de Usuario: HU3
Nombre Tarea: Implementar la funcionalidad para listar los roles del sistema.	
Tipo de Tarea: Desarrollo	
Fecha de inicio: 20-02-2014	Fecha de fin: 21-02-2014
Programador Responsable: Adonis Vázquez Campos	
Descripción: Se crea el código que permite listar un rol en el sistema.	

Tabla 10. Implementar la funcionalidad para listar los roles del sistema.



Tarea de ingeniería	
Número Tarea: 11	Número Historia de Usuario: HU3
Nombre Tarea: Implementar la funcionalidad para buscar un rol en el sistema.	
Tipo de Tarea: Desarrollo	
Fecha de inicio: 22-02-2014	Fecha de fin: 23-02-2014
Programador Responsable: Adonis Vázquez Campos	
Descripción: Se crea el código que permite buscar un rol en el sistema.	

Tabla 11. Implementar la funcionalidad para buscar un rol en el sistema.

Anexo 3. Casos de prueba de aceptación.

Caso de Prueba de Aceptación	
Código: HU2_P3	Historia de Usuario: 2
Nombre: gestionar usuario	
Nombre de la persona que realiza la prueba: Adonis Vázquez Campos	
Descripción: prueba para la funcionalidad de modificar los datos de un usuario en la aplicación.	
Condiciones de Ejecución: solo acceden a la información de los usuarios los administradores del sistema, de acuerdo al nivel del mismo.	
Entrada / Pasos de Ejecución: se intenta modificar los datos de un usuario.	
Resultado Esperado: se debe modificar los datos del usuario de manera correcta en la base de datos mostrando un mensaje de aviso e indicando que el proceso fue satisfactorio. En cualquier otro caso se debe mostrar un mensaje informándole sobre los errores en los que pudo incurrir y además comunicándole que no se pudo completar el proceso.	
Evaluación de la Prueba: prueba satisfactoria.	

Tabla 3.3. Prueba 2 al Módulo Gestionar Usuarios.



Caso de Prueba de Aceptación	
Código: HU2_P4	Historia de Usuario: 2
Nombre: gestionar usuario	
Nombre de la persona que realiza la prueba: Adonis Vázquez Campos	
Descripción: prueba para la funcionalidad de eliminar un usuario de la aplicación.	
Condiciones de Ejecución: solo acceden a la información de los usuarios los administradores del sistema, de acuerdo al nivel del mismo.	
Entrada / Pasos de Ejecución: se intenta eliminar los datos de un usuario.	
Resultado Esperado: se debe eliminar el usuario de manera correcta en la base de datos mostrando un mensaje de aviso e indicando que el proceso fue satisfactorio. En cualquier otro caso se debe mostrar un mensaje informándole sobre los errores en los que pudo incurrir y además comunicándole que no se pudo completar el proceso.	
Evaluación de la Prueba: prueba satisfactoria.	

Tabla 3.4. Prueba 3 al Módulo Gestionar Usuarios.

Caso de Prueba de Aceptación	
Código: HU2_P5	Historia de Usuario: 2
Nombre: gestionar usuario	
Nombre de la persona que realiza la prueba: Adonis Vázquez Campos	
Descripción: prueba para la funcionalidad de buscar un usuario en la aplicación.	
Condiciones de Ejecución: solo acceden a la información de los usuarios los administradores del sistema, de acuerdo al nivel del mismo.	
Entrada / Pasos de Ejecución: se intenta buscar un usuario en la aplicación.	



<p>Resultado Esperado: deben aparecer los datos del usuario de manera correcta, para su posterior modificación mostrando un mensaje de aviso e indicando que el proceso fue satisfactorio. En caso de que el administrador ignore algunos campos de carácter obligatorio o introduzca datos incorrectos se debe mostrar un mensaje informándole sobre los errores en los que pudo incurrir y además comunicándole que no se pudo completar el proceso.</p>
<p>Evaluación de la Prueba: prueba satisfactoria.</p>

Tabla 3.5. Prueba 4 al Módulo Gestionar Usuarios.

Caso de Prueba de Aceptación	
Código: HU2_P6	Historia de Usuario: 2
Nombre: gestionar usuario	
Nombre de la persona que realiza la prueba: Adonis Vázquez Campos	
Descripción: prueba para la funcionalidad de listar todos los usuarios de la aplicación.	
Condiciones de Ejecución: solo acceden a dicha información los administradores del sistema, de acuerdo al nivel del mismo.	
Entrada / Pasos de Ejecución: se intentan listar los usuarios.	
Resultado Esperado: debe aparecer la lista de todos los usuarios de manera correcta, para su posterior modificación mostrando un mensaje de aviso e indicando que el proceso fue satisfactorio. En caso contrario se debe mostrar un mensaje informándole sobre los errores en los que pudo incurrir y además comunicándole que no se pudo completar el proceso.	
Evaluación de la Prueba: prueba satisfactoria.	

Tabla 3.6. Prueba 5 al Módulo Gestionar Usuarios.