

Universidad Central “Marta Abreu” de las Villas.

Facultad Matemática Física y Computación

Ingeniería Informática.



TRABAJO DE DIPLOMA

Sitio web para el proyecto cultural “El Mejunje de Silverio”.

Autor: Alejandro Acosta Aguiar.

Tutor: Ing. Lianny Ofarrill Fernandez.

Consultante: Rolando Evelio Pérez Versón

“Año 54 de la Revolución”

Santa Clara

2012



Declaración Jurada

El que suscribe, _____, hago constar que el trabajo titulado _____ fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad de _____, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor

Firma del jefe del Laboratorio

Fecha

A los que le debo todo lo que soy: A mis padres por ser los mejores del mundo, por preocuparse tanto por mí, por todo el apoyo que siempre he podido encontrar en cada uno en todos los momentos de mi vida, porque sin importar las veces que caiga siempre han de estar para ayudar a levantarme. A mi hermano, a mi abuela y mi abuelo que son mis otros padres.

Alejandro.

Agradecimientos

A mis padres que a pesar de que logran estresarme han sido maravillosos y han confiado en mí todo este tiempo.

A mi hermano “la estrella de teatro” que a pesar de sus resabios lo quiero mucho.

A mi abuela que se la pasa peleando el día entero y a mi abuelo que es el viejo más lindo del mundo.

A Yaniel que se convirtió en mi hermano y que ha estado en las malas y las buenas a mi lado a pesar de sus cosas.

A mis niñas que las quiero mucho y aguantan mis cosas Yeila, Eliza y Dianita.

A mi hermana Betty que no está en Cuba y fueron muchas las notas juntos.

A los trabajadores de la facultad de MFC que me han malcriado por 5 años: Cristina, Migda, Yurien, Lucia, Anisley, Maribel, Sara, Rosi, Erenia, Oria, Anay, Dayana, Maida que solo ha sido un año.

A Lianna que tanto me ayudo con el diseño de la web.

A mis colegas del grupo que bastante nos jodimos unos a otros Aniel, Miguel, Joseito, Felix, Dustin, Ricardo.

A Yeidy, la Titi, Yuliet, el Yoa por ser especiales en mi vida y por llevarme a los campismos.

Al “El Mejunje” por inspirarme a hacer este trabajo y por no dejar que me aburriera en este camino tan largo y agotador.

A Silverio por ser el creador del mejunje y que ojala dejara entrar gratis a partir de ahora a mí y a mi tutora.

A mis profesores que han sido ejemplares.

A todos mis amigos y amigas que han estado a mi lado y como son muchos no los puedo mencionar a todos porque si no tendría alrededor de 10 páginas de agradecimientos.

Contenido

INTRODUCCIÓN.....	1
Antecedente	2
Hipótesis.....	3
Objetivo General	3
Objetivos específicos:	3
Tareas para darle cumplimiento a mis objetivos específicos.	4
Aportes prácticos esperados:	4
El informe del presente trabajo se ha dividido en tres capítulos:	4
Introducción	6
1.1 Objeto de estudio.....	6
1.1.1 Objetivo estratégico de la organización.	6
1.1.2 Flujo actual de los procesos.....	6
1.1.3 Análisis crítico de la ejecución de los procesos.....	8
1.1.4 Proceso objeto de automatización.....	8
1.1.5 Sistemas automatizados existentes vinculados al campo de acción	8
1.2 Tendencias y tecnologías actuales.....	9
1.2.1 Fundamentación del Entorno de Desarrollo, Lenguaje, Gestor de Base de Datos y Tecnología utilizados.	10
1.2.1.1 Lenguaje PHP.	10
1.2.1.2 Marcos de trabajo.	14
1.2.1.3 Manejadores de Bases de Datos.	18
1.2.1.4 Servidores web.....	20
1.2.1.5 Modelaciones de diagramas.....	22
1.2.1.6 Servicios Web.....	26
1.3 Lenguaje UML.....	29
Capítulo 2. “Propuesta de Solución”.	31
Introducción.	31

2.1 Diagrama AS-IS.....	31
2.2 Diagrama To-Be.	32
2.3 Requisitos del sistema.	32
2.3.1 Requisitos funcionales.....	32
2.3.2 Requisitos no funcionales.	33
2.4 Actores del sistema a automatizar.....	36
2.5 Diagrama de casos de uso del sistema.....	37
2.6 Nombre y descripción de los casos de uso del sistema.	38
2.7 Diagrama de clases del análisis.	47
2.8 Diagrama de colaboración.....	47
2.9 Diagrama de secuencia.	48
2.10 Diagrama Entidad-Relación.	49
2.11 Arquitectura M-V-C.....	49
2.12 Modelo de componente.....	53
2.13 Modelo de despliegue.	54
Capítulo 3. “Manual del usuario”.....	56
3.1 Manual de usuario para los servicios web.....	58
3.1.1 Procedimiento para consumirlo.	58
3.2 Evaluación de calidad web.	59
3.2.1 Importancia de hacer las pruebas de calidad.	59
3.2.2 Tipos de Pruebas de Software.....	59
3.3 Pruebas de calidad del Software hechas al sistema.	61
3.4 Estudio de factibilidad.....	63
3.4.1 Introducción.....	63
3.4.2 Planificación basada en casos de uso.....	63
3.4.3 Calculo de puntos de casos de uso ajustados.....	66
3.4.4 Calculo del esfuerzo.....	70

Tabla de contenido

3.5 Beneficios tangibles e intangibles.....	73
3.6 Análisis de costos y beneficios	73
3.7 Conclusiones.....	74
Conclusiones.....	75
Recomendaciones.	76
Referencias bibliograficas	77

INTRODUCCIÓN

Las Nuevas Tecnologías de la Información y las Comunicaciones (TIC) han posibilitado que la información se torne un recurso valioso por el que la sociedad compite, en tiempos antiguos no era posible la correcta y segura difusión de noticias por la lentitud con que se recibía la información, siempre llegaba tarde y no era oportuna, el análisis obtenido en un determinado día en realidad describía lo que había pasado una semana antes. Los avances actuales hacen posible capturar y utilizar la información en el momento que se genera.

El desarrollo de las ciencias informáticas ha constituido un gran salto evolutivo en las comunicaciones y medios de difusión. Se comienza a digitalizar la información impresa de manera que ésta y la información digital puedan coexistir por duplicado. El hombre no sólo puede documentarse muy fácil, también la aparición de nuevas tecnologías y sistemas de procesamiento le permite almacenar información de manera rápida y segura.

El centro cultural “El Mejunje de Silverio” es una institución que se dedica a divulgar y a fomentar la cultura en todas sus manifestaciones, acoge a las personas que de una forma u otra hacen arte y desean presentarlo al público. Este lugar es uno de los centros emblemáticos de la provincia ya que en él se acoge a todos los artistas que quiera ofrecer su arte ya sean profesionales o no. Es un lugar que alberga a todas las personas sin importar género, raza, nacionalidad, orientación sexual, forma de pensar en fin, música que escuche, en fin es el lugar en que la diversidad es lo principal que oferta. El director de este lugar tan diferente a logrado rescatar muchas de las raíces y tradiciones que se estaban olvidando y las ha introducido de forma que las personas que frecuentan el “local” se identifiquen con ellas de una forma u otra. Este centro está dividido en cuatro partes en las cuales se realizan diferentes actividades de formas simultáneas e independientes, estas divisiones son la Galería de Arte, la Sala de Teatro Margarita Casallas, el Patio y el Café. La galería de arte que acoge las exhibiciones de exposiciones de distintas manifestaciones de las artes plásticas así como de distintos temas. En la sala de teatro se planifican algunas actividades que se pueden realizar en lugares más pequeños como son la proyección de películas, las puestas en escena de

obras de teatro tanto de grupos de teatro profesionales como de grupos de teatro de aficionados y conciertos pequeños. El café es un espacio en el que se puede ir a conversar, tomar distintas infusiones, se ofrecen momentos en los que se puede interactuar con los protagonistas de algunas actividades realizadas por ejemplo la peña del tabaco, entre otras. El patio es el lugar donde en toda la semana se realiza alguna actividad tanto por las tardes como por la noche conjugando todos los géneros musicales. El Mejunje se concibe —dando crédito y razón a su nombre— como algo diverso e informal: alguien llega y dice un poema, se presenta un libro, se organiza un concierto de rock, los domingos muy temprano entran no menos de cien niños. Anualmente se reúne allí lo mejor del teatro de pequeño formato del país en un evento único. El mejunje además le abre las puertas a muchos festivales que se celebran en la ciudad y se convierte en una sede especial de la celebración, festivales como “Ciudad Metal”, “Festival de trova Longina” entre otros.

Antecedente

Como se evidencia en lo antes escrito el Mejunje es un centro al que muchas personas asisten no solo de la provincia sino de todo el país. Las personas que concurren a las actividades que allí se programan suelen **enterarse** por medio de las carteleras en las **mismas** puertas del centro, donde se ubican todas las actividades que se planifican en la semana dejando a suerte aquellas que aparecen de imprevisto, la radio ofrece la cartelera diaria a través de la emisora provincial quedando fuera de alcance el público de otras provincias del país, ocurriendo de la misma manera en el canal provincial “TeleCubanacán”, otra vía es por medio de las personas que frecuentan el lugar y divulgan de manera informal las actividades. Hoy en día estos medios no bastan para difundir toda la información necesaria con respecto a la vida diaria del local ya que no llegan de manera estable a una buena parte del público interesado.

El centro cultural “El Mejunje de Silverio” **dispone de un sitio Web** el cual está disponible en un servidor fuera del país, lo que hace que sea accesible por un número restringidos de usuarios dentro de las redes nacionales. El mantenimiento y la actualización del sitio se dificultan debido a que los administradores no tienen acceso de forma constante a los sucesos que ocurren de manera frecuente o imprevista en el complejo cultural. Las funcionalidades que brinda el sistema se limitan a la presentación

de información lo que hace que el flujo de comunicación sea en un solo sentido, de manera que no se pueda interactuar eficientemente con el sitio ni con otros usuarios. La interfaz de usuario no es amigable, la información no presenta una distribución adecuada ya que están disponibles todas las actividades de manera secuencial independientemente del lugar, el tipo de actividad y la fecha provocando desorganización y que el usuario consulte obligatoriamente información no deseada.

Es por eso que se ha estado pensando hacer un sitio en el que la información esté mejor distribuida, que las noticias y actividades de último momento sean actualizadas para que pueda llegar a tiempo a una parte de las personas.

A partir de lo antes expuesto se plantea el siguiente **problema científico**:

La no existencia de un medio que permita de manera integradora e interactiva la comunicación e interacción constante entre el Mejunje de Silverio, el centro provincial de cultura, todas las personas y entidades interesadas en el complejo cultural dentro y fuera de la isla.

Hipótesis

Si se implanta un sistema Web al cual tengan acceso la mayoría de las personas y entidades dentro y fuera de la isla se puede lograr la comunicación entre el “Mejunje de Silverio”, el centro provincial de cultura, y todas las personas interesadas en el complejo cultural de forma más interactiva y eficiente.

Objetivo General

Implementar un sistema Web al cual tengan acceso la mayoría de las personas y entidades dentro y fuera de la isla para lograr la comunicación entre el “Mejunje de Silverio”, el centro provincial de cultura, y todas las personas interesadas en el complejo cultural de forma más interactiva y eficiente.

Objetivos específicos:

- Realizar la captura de los requerimientos del sistema a desarrollar.
- Diseñar una Base de datos.
- Implementar un servicio web que brinde información relacionada con las actividades que acontecen en el centro.
- Desarrollar un sitio web que responda a los requerimientos de los usuarios.
- Realizar pruebas de calidad al producto.

- Implantar el sistema.

Tareas para darle cumplimiento a mis objetivos específicos.

- Estudiar cómo se realiza el proceso de publicar las actividades que se hacen en el centro para ponerlas en la cartelera, con el objetivo de encontrar mecanismos que apoyen la realización de estas actividades y puedan llevarse a cabo de una manera más eficiente.
- Identificar el tipo de información que es necesaria incluir en el sistema.
- Almacenar y mostrar la información que pueda ser de interés público y los datos que resulten importantes para la entidad.
- Estudiar las tecnologías que mundialmente son empleadas en la actualidad para el desarrollo de Sistemas web.
- Efectuar reuniones con el personal especializado y encargado de gestionar las carteleras en el centro cultural.
- Realizar un diseño sencillo y agradable para los usuarios del sistema que permita la rápida fluidez de la información desde los servidores hasta los navegadores de las computadoras y el fácil manejo para los usuarios.
- Diseñar la Base de Datos que contendrá la información requerida.
- Desarrollar una adecuada Ingeniería de Software que permita cumplir con todas las funcionalidades esperadas.

Aportes prácticos esperados:

- Brindar un sistema que tenga en cuenta los requerimientos de sus usuarios.
- Automatizar la publicación de la cartelera cultural del centro y la publicación de artículos de interés sobre la vida del centro.

El informe del presente trabajo se ha dividido en tres capítulos:

Capítulo 1. “Las nuevas tecnologías de la información y las comunicaciones en la divulgación de información “, este capítulo describe el marco teórico del tema y el estado del arte en el plano internacional y Cuba.

Capítulo 2. “Análisis y diseño del sistema”, en este segundo capítulo se tratan aspectos propios del análisis y diseño sistema que se va a desarrollar así como un análisis de la tecnología que se va a utilizar para su implementación y desarrollo.

Capítulo 3. “Manual del usuario”, está destinado a ofrecer una breve guía o manual para el trabajo de los diferentes usuarios del sistema. Pruebas de calidad hechas a sistema.

Introducción.

En el presente capítulo se hace una descripción general del objeto de estudio y se especifica cómo se desenvuelven los procesos que tienen que ver con el manejo y mejoramiento de las publicaciones en el Centro Cultural el Mejunje, tanto de las actividades que se realizan como de los artículos referentes a los intereses del centro. Además se dará a conocer la necesidad de construir un sitio web para el complejo cultural y se realizará un estudio de las tecnologías de desarrollo a utilizar.

1.1 Objeto de estudio.

El objeto de estudio comprende toda la investigación relacionada con el ambiente que rodea al sistema que se desea implantar, que abarca desde la vida interna del Mejunje, la repercusión que tiene el centro en la sociedad cubana y extranjera, hasta el estudio de las posibles herramientas y metodologías que se deben usar.

1.1.1 Objetivo estratégico de la organización.

El centro cultural “El Mejunje de Silverio” es un lugar de gran prestigio en el país por su diversidad cultural de gran radio. Este lugar se encarga totalmente de difundir la cultura. Es un territorio, por decirlo de algún modo, en el cual se acoge a todo tipo de personas, con forma de pensar, gustos, preferencias musicales y cualquier cosa diferente y sin embargo se crea un ambiente acogedor. Es un lugar creado para rescatar argumentos de nuestras raíces, para el encuentro con valores de la cultura no solo nacional sino de la localidad, para desarrollar en las personas que allí asisten nuevas formas de pensar sobre todo en el público más joven.

1.1.2 Flujo actual de los procesos.

El centro cultural “El mejunje de Silverio” cuenta actualmente con cuatro lugares en los que se realizan, de forma simultánea o no, diferentes actividades. Ideas para actividades nuevas nunca faltan. Todos los días de la semana se realizan diferentes actividades a distintas horas, a veces hay actividades en la sala de teatro y hasta que estas no terminen no comienzan las actividades programadas para el patio. Todo se sincroniza y se maneja de una forma organizada para que no existan contratiempos.

- La sala de teatro acoge tanto aficionados como a profesionales de las artes escénicas. Es un escenario donde muchos grupos de teatro han querido estrenar sus obras por el tipo de público que frecuenta este espacio. Es el lugar donde se hacen conciertos de pequeño formato, se proyectan películas que han marcado la historia del cine y para muchas actividades que requieran un ambiente como el de la “salita”.
- El Café-Mejunje es el lugar donde ofertan variedades de cafés. Muchas veces en este local se hacen actividades como la peña del tabaco, El Guayabazo donde los protagonistas son las personas que allí asisten junto con un grupo musical que hace música tradicional cubana, se proyectan videos musicales antiguos y otras.
- La galería de arte es el espacio en que se hacen exposiciones de todas las manifestaciones de las artes plásticas, una exposición se mantiene durante un mes. Además en varias ocasiones se hacen exposiciones en honor a alguna personalidad del arte, sobre la vida de algún grupo de teatro o de cualquier tema que sea con respecto a algo.
- El patio, el más grande de los espacios del mejunje y el que en la vida cultural es que el mayor protagonismo tiene. Aquí están las actividades permanentes que se realizan de forma diaria. Cada día de la semana el patio abre las puertas con una peña diferente y una propuesta para todos los gustos, trova, música electrónica, rock and roll, música tradicional cubana, espacios para los niños. Es en donde la mayoría de las actividades imprevistas se realizan conciertos, obras de teatro que requieren de un espacio más grande, además de las grandes mayorías de los festivales organizados por el centro.

Actualmente el centro maneja diferentes maneras de divulgar la información:

- La publicación de una cartelera en sus puertas con las actividades a realizar en la semana,
- La divulgación de las actividades más importante en los medios de difusión locales prensa, radio y televisión.
- El intercambio de información por parte de la dirección del complejo cultural a través de correos electrónicos con diferentes usuarios.

1.1.3 Análisis crítico de la ejecución de los procesos.

Las formas de divulgación citadas anteriormente ayudan a la propaganda de las actividades, pero no logran abarcar a todo el personal que puede estar interesado en las propuestas del Mejunje. En el caso de la cartelera al estar situada en las puertas del centro solo es visible para aquellos que pasen por su dirección, los medios de difusión por su parte a pesar del gran esfuerzo que realizan, por su condición de locales son de alcance provincial y solamente se encargan de las actividades más importantes, dejando fuera muchas personas interesadas de conocer lo que pasa en la casa de Silverio y sin proponer algunas de las actividades que allí se cometen.

A pesar de que la dirección de cultura de la ciudad de Santa Clara hace un esfuerzo por mantener al tanto a la población de la vida del Mejunje aún es insuficiente la divulgación de su cartelera semanal y diaria.

1.1.4 Proceso objeto de automatización.

- Brindar un servicio que sea atractivo y fácil de usar para las personas que lo visiten.
- Gestionar la información de las actividades de las carteleras para las diferentes áreas.
- Tratar artículos y materiales de interés del centro.

1.1.5 Sistemas automatizados existentes vinculados al campo de acción

Actualmente para manejar la información referente al centro cultural “El Mejunje de Silverio” existe un sitio publicado, que no es oficial, en un servidor extranjero, el cual es un sitio que solo brinda información, no permite la retroalimentación con los usuarios. Existen varias razones que respaldan la conclusión de que este sistema no brinda las funcionalidades necesarias para toda la información referente a la vida cultural del centro:

- Al estar el sitio en un servidor fuera del país, la actualización de la información tanto de los artículos que se puedan publicar como de las actividades que se realizan en cualquiera de las áreas del centro se hace prácticamente imposible.
- Los usuarios que quieran acceder al sitio tienen que tener internet para poder ver la información.
- Los usuarios no pueden dar su opinión sobre uno de los artículos publicado.
- No se actualizan las actividades que aparecen a última hora de improviso.

Vale aclarar que el sitio, además de presentar las dificultades e inconvenientes antes mencionados en la actualidad no se encuentra en funcionamiento.

1.2 Tendencias y tecnologías actuales.

Dado el hecho de que el sistema que se necesita debe ser usado por múltiples usuarios concurrentemente, y mantener sus datos actualizados y disponibles a tiempo completo se decidió desarrollar una aplicación Web, para dar respuestas a las necesidades.

Las aplicaciones Web son populares debido a la practicidad del navegador Web como cliente ligero.

La facilidad de uso de un entorno Web está completamente relacionada con el diseño del mismo. El sitio debe estar construido de forma tal que los usuarios sepan que hacer en cualquier momento y cómo hacerlo, lo cual facilitaría el aprendizaje en el menor tiempo posible de sus características.

Inmediatamente el visitante al acceder al sitio debe conocer los objetivos y alcance del mismo. Suele ayudar una distribución y diseño gráfico adecuados así como un lenguaje claro e inteligible para comunicarlo lo que influye en el usuario positivamente, promoviéndolo a permanecer en el entorno Web e incitándolo a volver a acceder al mismo. Es importante que los mejores contenidos o los que puedan despertar mayor interés queden expuestos de manera visible para que el visitante conozca de su presencia rápidamente contribuyendo a que se adquiera una visión de sitio valioso. (Keeker, 2009)

La Web ha permitido un flujo de comunicación global a una escala sin precedentes en la historia humana. Personas separadas en el tiempo y el espacio, pueden usar la Web para intercambiar- o incluso desarrollar mutuamente- sus pensamientos más íntimos, o alternativamente sus actitudes y deseos cotidianos. Experiencias emocionales, ideas políticas, cultura, idiomas musicales, negocio, arte, fotografías, literatura... todo puede ser compartido y diseminado digitalmente con el menor esfuerzo, haciéndolo llegar casi de forma inmediata a cualquier otro punto del planeta. Aunque la existencia y uso de la Web se basa en tecnología material, que tiene su vez sus propias desventajas, esta información no utiliza recursos físicos como las bibliotecas o la prensa escrita. Sin embargo, la propagación de información a través de la Web no está limitada por el

movimiento de volúmenes físicos, o por copias manuales o materiales de información. Gracias a su carácter virtual, la información en la Web puede ser buscada más fácil y eficientemente que en cualquier medio físico, y mucho más rápido de lo que una persona podría recabar por sí misma a través de un viaje, correo, teléfono, telégrafo, o cualquier otro medio de comunicación.

1.2.1 Fundamentación del Entorno de Desarrollo, Lenguaje, Gestor de Base de Datos y Tecnología utilizados.

Después de un estudio profundo se llegó a la conclusión que entre los lenguajes y tecnologías más difundidos actualmente en el mundo de la Web se encuentran: Ap.Net, PHP, ColdFusion y Java. Estos se caracterizan por llevar la lógica del negocio dentro del servidor, además de ser los encargados del acceso a bases de datos, tratamiento de la información, etc. También se valoraron diferentes gestores de bases de datos, entre ellos: Postgre, MySQL, SQL Server 2000 y Oracle.

1.2.1.2 Lenguaje PHP.

PHP es un lenguaje interpretado de propósito general ampliamente usado, diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Pearl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, flash, así como imágenes en diferentes formatos.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite. Además PHP tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX (y de ese tipo, como Linux o Mac OS X) y Windows, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

PHP es una alternativa a las tecnologías de Microsoft ASP y ASP.NET (que utiliza C# VB.NET como lenguajes), a ColdFusion de la compañía Adobe (antes Macromedia), a JSP/Java de Sun Microsystems, y a CGI/Perl. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un IDE (Entorno de Desarrollo Integrado) comercial llamado Zend Studio. Recientemente, CodeGear (la división de lenguajes de programación de Borland) ha sacado al mercado un IDE para PHP, denominado **Delphi for PHP**. Existe un módulo para Eclipse, uno de los IDE más populares. (Ecured, 2009b)

Ventajas

- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial (Sitio Oficial), entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.

- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño **Modelo Vista Controlador** (MVC), que permiten separar el tratamiento y acceso a los Datos, la Lógica de control y la Interfaz de usuario en tres componentes independientes.

Desventajas.

- La Ofuscación de código es la única forma de ocultar las fuentes.

Versiones y características de PHP. (wikipedia, 2011b)

Versión	Cambios más importantes.
PHP 1.0	Oficialmente llamado "Herramientas personales de trabajo (PHP Tools)". Es el primer uso del nombre "PHP".
PHP 2	Considerado por el creador como la "más rápida y simple herramienta" para la creación de páginas webs dinámicas.
PHP 3.0	Desarrollo movido de una persona a muchos desarrolladores. ZeevSuraski y AndiGutmans reescriben la base para esta versión.
PHP 4.0	Se añade un sistema más avanzado de análisis de etiquetas en dos fases análisis/ejecución llamado el motor Zend.
PHP 4.1	Introducidas las variables superglobals (\$_GET, \$_SESSION, etc.).
PHP 4.2	Se deshabilitan register_globals por defecto.
PHP 4.3	Introducido la CLI, en adición a la CGI.
PHP 4.4	
PHP 5.0	Motor Zend II con un nuevo modelo de objetos.
PHP 5.1	
PHP 5.2	Habilitado el filtro de extensiones por defecto.
PHP 5.2.4	

PHP 5.2.5	Versión centrada en mejorar la estabilidad (+60 errores solucionados).
PHP 5.2.8	
PHP 5.2.9	Diversas mejoras en el ámbito de la seguridad (+50 errores solucionados).
PHP 5.2.12	Diversas mejoras en el ámbito de la seguridad (+50 errores solucionados).
PHP 5.3	Se le agrega namespaces, late static binding, closures, optional garbage collection for cyclic references, nuevas extensiones (+140 errores solucionados).
PHP 5.3.1	Diversas mejoras en el ámbito de la seguridad (36 errores solucionados).
PHP 5.3.2	Diversas mejoras en el ámbito de la seguridad (99 errores solucionados).

Marcos de trabajos que usan PHP.

- Zend Framework
- Kohana
- Symfony
- CakePHP
- PHP Prado
- CodeIgniter
- Lithium
- Sapphire

El uso del lenguaje PHP es utilizado en distintos marcos de trabajos y la conformación de sitios web dinámicos a continuación se hace un análisis de alguno de los marcos de trabajos usados hoy en día.

1.2.1.3 Marcos de trabajo.

CakePHP

CakePHP, es un marco de trabajo para programar aplicaciones web que sigue la arquitectura Modelo, Vista, Controlador (MVC). Es una estructura de librerías, clases y estructuras run-time para programadores que creaban aplicaciones web originalmente inspiradas en el marco de trabajo *Ruby onRails*. O sea, se trata de un esqueleto o andamiaje que sirve entre otras cosas para crear aplicaciones web con php. Hay que definir una base de datos, unas tablas, a partir de ahí se definen las vistas de cómo mostrarlo y con unos controladores que nos sirven para la creación de eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. Su principal meta es permitir el trabajo de manera estructurada y rápida, sin pérdida de flexibilidad. <17>

Características de CakePHP.

CakePHP posee varias características que lo hacen que sea una gran elección como plataforma para el desarrollo de aplicaciones rápidamente y con la menor cantidad de molestia. La principal de ellas es el lenguaje de programación en que se basa, PHP, que a su vez es el más usado en entornos web. Además, también hay que destacar su activa y colaborativa comunidad, que no se limita solo al sitio web oficial de cake, sino que también hace posible la existencia de repositorios de aplicaciones completas hechas con CakePHP, la publicación de pedazos de códigos que pueden ser útiles para otros usuarios, o un sistema de control de errores donde se pueden reportar y/o comprobar las respuestas a ellos, y así sería una vía de buscar ayuda cuando se necesite. CakePHP posee una licencia muy flexible, ya que al usar la licencia MIT, nos permite no solo usar, copiar o modificar el código, sino publicar, sub-licenciar o vender copias de nuestra aplicación desarrollada con esta plataforma; compatibilidad con PHP4 y PHP5, que nos permitirá aprovecharnos de las nuevas mejoras que incorpora PHP5, además de instalar y usar CakePHP en hosting que solo nos ofrezcan PHP4. CakePHP incorpora además una característica muy útil, especialmente en el periodo de puesta en marcha de una aplicación, el *Scaffolding*. Si activamos el *Scaffolding* para una aplicación, o para una parte de ella, es el propio código de CakePHP el que se encargara de analizar la estructura de la base de datos y crear listados de los registros con botones para insertar

nuevos registros, borrar o editar los existentes. Asimismo también crea los formularios para añadir o editar los registros. El *Scaffolding* no es completamente flexible, por lo que posiblemente si nuestra aplicación requiere de un cierto grado de complejidad, deberemos desactivarlo y crear nosotros mismos los formularios, aunque, en la fase inicial del proyecto, en la que una estructura de base de datos aún pueda sufrir cambios, resulta un método muy potente y sobre todo fácil y rápido para probar el diseño. También incluye una serie de librerías con funciones de AJAX, *Javascript*, Formularios HTML, etc., que permitirá y se encargará, además de desarrollar más rápidamente, de tener un código completamente basado en la nomenclatura de CakePHP sin necesidad de mezclar distintos tipos de lenguajes. Usa una arquitectura MVC, que obliga a seguir una estructura determinada, de la que se tendrá el beneficio a la hora de mantener o ampliar las funcionalidades de nuestro código. (Ramirez, 2010)

Estructura de CakePHP

CakePHP utiliza el modelo vista-controlador que es un patrón para diseño de software que ayuda a separar el código haciéndolo reusable, mantenible y generalmente mejor. La entrada del usuario, el modelado del mundo externo, y la retroalimentación visual para el usuario son separados y dirigidos por los objetos modelo, vista y controlador.

En términos de Cake, el Modelo representa una base de datos, tabla o registro, así como sus relaciones con otras tablas o registros. Los modelos contienen las reglas para la validación de datos. Estas reglas son aplicadas cuando en el modelo se insertan o actualizan datos. Las vistas en Cake están representadas por los archivos “*views*”, los cuales son archivos HTML con códigos PHP incluido. Los controladores manejan las peticiones del servidor. Toma las entradas del usuario (URL y datos de Post), aplica la lógica de negocio, utiliza los modelos para leer y escribir en una base de datos, y manda la salida apropiada de datos con los archivos de vistas. (Ramirez, 2010)

Requerimientos para la instalación.

Para usar CakePHP se debe tener primero un servidor que tenga las librerías y programas para correr CakePHP.

1. Un servidor HTTP (como Apache) habilitado con: sesiones.
2. PHP 4.3.2 o superior.

3. Un motor de Base de datos (ya sea MySQL, PostgreSQL y un envoltorio para ADODB).

Facilidades de CakePHP.

CakePHP facilita la tarea de crear las aplicaciones, pero también conseguirá dotar de una buena metodología a la hora de programar, lo que incentiva la reutilización del código en futuras aplicaciones. CakePHP usa un patrón de arquitectura de *software* denominado Modelo Vista Controlador que obliga, aunque sin dejar de ser flexible, a separar los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. Es decir, por un lado se definirá el Modelo, que básicamente será la conexión con una tabla de la base de datos aunque CakePHP ofrece potentes funciones con las que se pueden definir relaciones entre distintos Modelos (tablas). Por otro lado se tiene el Controlador, que será donde se incluirán la lógica de la aplicación, es decir, se definirán los métodos que interactuarán con los datos (Modelo) y mostrarán el resultado en el navegador usando las Vistas. Las acciones que realiza un usuario de una aplicación serían posibles llamadas a estos métodos del Controlador. Las Vistas serán plantillas HTML con partes de código CakePHP incluidos. (Ramirez, 2010)

Symfony.

Symfony es un marco de trabajo diseñado para optimizar el desarrollo de aplicaciones web a través de diversas características clave. Separa las reglas de negocio de la aplicación, la lógica del servidor y las vistas de presentación. Contiene una gran variedad de herramientas y clases para conseguir acortar el tiempo de desarrollo de aplicaciones web complejas. Adicionalmente, automatiza tareas comunes para que el programador pueda enfocarse por completo en las especificaciones. Fácil de instalar y configurar: ha sido probado con éxito en plataformas Windows y derivadas de Unix. Cumple con la mayoría de las mejores prácticas en diseño web y patrones de diseño.

Zend.

Zend es un marco de trabajo que destaca el hecho de que no sólo busca facilitar la programación a través del patrón MVC, sino también automatizar tareas más específicas, como el acceso a base de datos, el filtrado de datos ingresados a la aplicación o la búsqueda en un sitio web ordenando resultados por relevancia. Permite

que la aplicación web pueda hacer uso de servicios web ofrecidos por aplicaciones externas y exponga servicios web propios. Intenta facilitar la lectura del código fuente, brindar ejemplos de su uso, promover equipos de traducción de la documentación y elaborar un tutorial del desarrollo de una aplicación web.

Cuadro Comparativo de marcos de trabajo en PHP.

Características	CakePHP	Symfony	Zend
1-Arquitectura de aplicaciones			
Incorporación del patrón Modelo Vista Controlador orientado a objetos.	X	X	X
Operaciones CRUD (Create, Retrieve, Update y Delete) asociadas a patrón Active Record.	X	X	
Mapeado de objetos a bases de datos relacionales (ORM).	X	X	
Independiente del manejador de base de datos.	X	X	X
Estructura por defecto para aplicaciones (scaffolding).	X	X	
Archivos de configuración de la aplicación.	.php	.yaml*	.php
2-Acceso vía web			
Despachador de peticiones HTTP.	X	X	X
Generación de URLs amigables.	X	X	X
3-Implementación de código HTML			
Uso de plantillas en PHP.	X	X	X
Posibilidad de uso de plantillas en Smarty.	X	X	
Implementación de ayudantes de plantillas (helpers).	X	X	
4-Seguridad			
Manejo propio de sesiones por usuarios.	X	X	X
Manejo de privilegios de acceso a secciones de la aplicación.	X		X

Verificación de la salida generada en HTML por procesamiento de peticiones.	X	X	
5-Usabilidad y acceso rápido			
Almacenamiento en caché de las vistas.	X	X	
6-Herramientas de programación			
Generación de código PHP.	X	X	
Interfaz de línea de comandos para la creación y mantenimiento de aplicaciones.	X	X	
7-Extensibilidad y opciones adicionales			
Integración con otras herramientas a través de plugins.	X	X	
Implementación propia de llamadas Asynchronous JavaScript and XML (AJAX)	X	X	
8-Soporte PHP			
Soporte para PHP4.	X		
Soporte para PHP5.	X	X	X

Todos estos marcos de trabajos utilizan distintos manejadores de bases de datos para el uso de las mismas.

1.2.1.4 Manejadores de Bases de Datos.

MySQL.

MySQL es la base de datos de código abierto más popular del mundo. Su continuo desarrollo y su creciente popularidad están haciendo de MySQL un competidor cada vez más directo de gigantes en la materia de las bases de datos. MySQL es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales.

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad relacional y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad.

Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores del software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferente velocidad de operación, soporte físico, capacidad, distribución geográfica, transacciones...
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.
- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
- Soporta gran cantidad de tipos de datos para las columnas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.
- Infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación.
- Fácil instalación y configuración.

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por

relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.
(wikipedia, 2011d)

SQL Server 2005 Express

SQL Server 2005 Express es una plataforma global de base de datos que ofrece administración de datos con herramientas integradas de inteligencia empresarial (BI). El motor de la base de datos SQL Server 2005 Express ofrece un almacenamiento seguro y confiable tanto para datos relacionales como estructurados.

El motor de datos SQL Server 2005 Express constituye el núcleo de esta solución de administración de datos. Asimismo, SQL Server 2005 combina lo mejor en análisis, información, integración y notificación. Esto permite que su sistema cree y despliegue soluciones de BI rentables que ayuden a su equipo a incorporar datos en cada rincón del negocio a través de tableros de comando, escritorios digitales, servicios Web y dispositivos móviles. (Jacobson)

Al trabajar con un manejador de Base de Datos se necesitan herramientas que te permitan la instalación y ejecución de dicho manejador, para eso se pueden utilizar varias tecnologías que permiten el montaje de un sistema web.

1.2.1.5 Servidores web.

XAMPP.

XAMPP, es un servidor de plataforma libre, es un software que integra en una sola aplicación, un servidor web Apache, intérpretes de lenguaje de scripts PHP, un servidor de base de datos MySQL, un servidor de FTP FileZilla, el popular administrador de base de datos escrito en PHP, MySQL, entre otros módulos. Te permite instalar de forma sencilla Apache en tu propio ordenador, sin importar tu sistema operativo (Linux, Windows, MAC o Solaris). Y lo mejor de todo es que su uso es gratuito. El programa está liberado bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y MacOS X.

Oficialmente, los diseñadores de XAMPP sólo pretendían su uso como una herramienta de desarrollo, para permitir a los diseñadores de sitios webs y programadores testear su trabajo en sus propios ordenadores sin ningún acceso a Internet. En la práctica, sin

embargo, XAMPP es utilizado actualmente como servidor de sitios Web, ya que, con algunas modificaciones, es generalmente lo suficientemente seguro para serlo. Con el paquete se incluye una herramienta especial para proteger fácilmente las partes más importantes.

XAMPP es una herramienta de desarrollo que te permite probar tu trabajo (páginas web o programación por ejemplo) en un propio ordenador sin necesidad de tener que conectarse a internet.

Si un desarrollador que recién está comenzando, XAMPP provee de una configuración totalmente funcional desde el momento que se instala sin embargo, es bueno acotar que la seguridad de datos no es su punto fuerte, por lo cual no es suficientemente seguro para ambientes grandes o de producción.

XAMPP es un paquete formado por un servidor web Apache, una base de datos MySQL y los intérpretes para los lenguajes PHP y Perl.

Características de Xampp.

- Para Windows existen dos versiones, una con instalador y otra portable (comprimida) para descomprimir y ejecutar.
- Otra característica no menos importante, es que la licencia de esta aplicación es GNU, está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.
- La filosofía de XAMPP, como lo indican en su sitio web, es crear una distribución fácil de instalar, de tal manera que los desarrolladores web principiantes cuenten con todo lo necesario ya configurado.
- XAMPP solamente requiere descargar y ejecutar un archivo .zip, .tar, o .exe, con unas pequeñas configuraciones en alguno de sus componentes que el servidor Web necesitará. XAMPP.
- Una de las características sobresalientes de este sistema es que es multiplataforma, es decir, existen versiones para diferentes sistemas operativos, tales como: Microsoft Windows, GNU/Linux, Solaris, y MacOS X. Existen versiones para Linux (testado para SuSE, RedHat, Mandrake y Debian),

Windows (Windows 98, NT, 2000, XP y Vista), MacOS X y Solaris (desarrollada y probada con Solaris 8, probada con Solaris 9).

Ventajas.

- Xampp es una herramienta muy práctica que nos permite instalar el entorno MySQL, Apache y PHP, suficiente para empezar proyectos web o revisar alguna aplicación localmente. Además trae otros servicios como servidor de correos y servidor FTP.
- Su instalación es sencilla, basta descargarlo, extraerlo y comenzar a usarlo. En general es bastante fácil la instalación de apache y php sobre Unix, sobre todo si dispone de un manejador de paquetes.
- Las configuraciones son mínimas o inexistentes, lo cual ahorra bastante tiempo.

Desventajas.

- Xampp trae las últimas versiones de las aplicaciones que instala, sin embargo cuando pasa el tiempo y salen nuevas versiones de las mismas, no queda otra salida que reinstalar todo Xampp.
- Este programa se distribuye con la esperanza de que sea útil, mas sin ninguna garantía, sin siquiera la garantía implícita de su mercadibilidad y aptitud para un propósito determinado.
- Xampp trae PhpMyAdmin para administrar las bases de datos de MySQL, sin embargo para tareas más específicas es mejor utilizar la consola y Xampp no la soporta.

1.2.1.6 Modelaciones de diagramas.

Herramientas CASE (Cumputer Aided Software Engineering). Conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de Software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software. Este puede ser generalmente aplicado a cualquier sistema o colección de herramientas que ayudan a automatizar el proceso de diseño y desarrollo de software.

Objetivos.

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software.

- Reducir el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto
- Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
- Automatizar el desarrollo del software, la documentación, la generación de código, las pruebas de errores y la gestión del proyecto.
- Ayuda a la reutilización del software, portabilidad y estandarización de la documentación
- Gestión global en todas las fases de desarrollo de software con una misma herramienta.
- Facilitar el uso de las distintas metodologías propias de la ingeniería del software.

Visual Paradigm.

Visual Paradigmas una de las herramientas UML CASE del mercado, considerada como muy completa y fácil de usar, con soporte multiplataforma y que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones.

Fue creada para el ciclo vital completo del desarrollo de software que lo automatiza y acelera, permitiendo la captura de requisitos, análisis, diseño e implementación. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de las clases.

Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación. Está diseñada para usuarios interesados en sistemas de software de gran escala con el uso del acercamiento orientado a objeto, además apoya los estándares más recientes de las notaciones de Java y de UML. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros.

Características

- Producto de calidad

- Soporta aplicaciones Web.
- Varios idiomas.
- Generación de código para Java y exportación como HTML.
- Fácil de instalar y actualizar.
- Compatibilidad entre ediciones.
- Se integra con las siguientes herramientas Java:
 - Eclipse/IBM WebSphere.
 - Jbuilder.
 - NetBeans IDE.
 - Oracle Jdeveloper.
 - BEA Weblogic.

Ventajas

- Apoya todo lo básico en cuanto a artefactos generados en las etapas de definición de requerimientos y de especificación de componentes.
- Tiene apoyo adicional en cuanto a generación de artefactos automáticamente.
- Genera modelos VP-UML instantáneamente a partir de código binario .Net.
- Generación de documentación en formatos HTML y PDF.
- Disponibilidad en múltiples plataformas: Microsoft Windows (98, 2000, XP, o Vista), Linux, Mac OS X, Solaris o Java.
- Brinda la posibilidad de intercambiar información mediante la importación y exportación de ficheros con aplicaciones como por ejemplo Visio y Rational Rose.
- Generación de código e ingeniería inversa: brinda la posibilidad de generar código a partir de los diagramas, para las plataformas como .Net, Java y PHP, así como obtener los diagramas a partir del código.
- Generación de documentación: brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas.

Desventajas

- Las imágenes y reportes generados, no son de muy buena calidad.

Tipos de diagramas que se pueden conformar.

Representación gráfica de una colección de elementos de modelado, a menudo dibujada como un grafo con vértices conectados por arcos. Diferentes tipos de diagrama permiten ver el sistema desde diferentes perspectivas. Los tipos de diagramas que se pueden incluir son: (Viscaino, 2008)

- Diagramas de Componentes: Describen la organización de los elementos físicos que implementan el sistema.
- Diagrama de Despliegue: Describen la configuración del entorno de máquinas y redes sobre el que se distribuyen componentes y procesos del sistema.
- Diagrama de Secuencia: Describen la interacción entre elementos del sistema en el tiempo.
- Diagramas Casos de Uso: Representan la funcionalidad del sistema.
- Diagrama de Clase: Describen la estructura (estática) del sistema.
- Diagrama de Actividad: Describen cómo se desarrolla un flujo de actividades entre elementos del sistema o del dominio.
- Diagrama de Estado: Describen el estado, condiciones y respuesta de los elementos del sistema.

POSEIDON

Es una herramienta para modelar cualquier clase de sistema que esté o no relacionada con programación. Poseidon para UML puede simplificar la compleja tarea de desarrollo de software ayudando a estructurar pensamientos, a clarificar la comunicación, y a encontrar la correcta abstracción. La incorrecta implantación de la herramienta UML, le sumergirá en detalles llenos de funciones extrañas y excesivamente complicadas, lo que le evitará el ahorro de tiempo y esfuerzo.

La intuitiva interfaz hace de Poseidon una de las herramientas más rápidas de UML para dominar el análisis orientado a objetos, liberando al diseñador para centrarse solamente en su modelo. (Ecured, 2011)

Desventajas

- En la versión Trial la grabación de proyectos está limitada a ocho diagramas.

Argo

Es una aplicación de diagramado de UML escrita en Java y publicada bajo la Licencia BSD (Berkley Software Distribution) código abierto. Dado que es una aplicación Java, está disponible en cualquier plataforma soportada por Java.

Sin embargo, desde la versión 0.20, ArgoUML está incompleto. No es conforme completamente a los estándares UML y carece de soporte completo para algunos tipos de diagramas de secuencia y los de colaboración. (Ecured, 2011)

Desventajas

- Instalación costosa.
- Poco amigable.
- Difícil de empezar.
- No tiene botón "deshacer".
- Los modelos a veces no pueden ser re-abiertos.
- No hay llamadas reflexivas en los diagramas de secuencia.
- Se debe seleccionar una clase para crear un diagrama de secuencia.

1.2.1.7 Servicios Web

Un **servicio web** (en inglés, *Web Service*) es una pieza de software que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares. (Wikipedia, 2011c)

Ventajas.

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.

- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.

Desventajas.

- Para realizar transacciones no pueden compararse en su grado de desarrollo con los estándares abiertos de computación distribuida como CORBA (CommonObjectRequestBrokerArchitecture).
- Su rendimiento es bajo si se compara con otros modelos de computación distribuida, tales como RMI (RemoteMethodInvocation), CORBA o DCOM (DistributedComponentObjectModel). Es uno de los inconvenientes derivados de adoptar un formato basado en texto. Y es que entre los objetivos de XML no se encuentra la concisión ni la eficacia de procesamiento.
- Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en *firewall* cuyas reglas tratan de bloquear o auditar la comunicación entre programas a ambos lados de la barrera.

SOAP

SOAP (siglas de *Simple Object Access Protocol*) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP fue creado por Microsoft, IBM y otros y está actualmente bajo el auspicio de la W3C. Es uno de los protocolos utilizados en los servicios Web. (Gomez, 2010)

La idea detrás de SOAP es la misma que RPC. También define un protocolo para llamadas a métodos remotos, sin embargo SOAP contiene (Fountes, 2008):

- Información adicional incluida en el documento XML (lenguaje de marcado extensible), que describe el contenido y como podría ser procesada.
- Definición de la especificación de algunas estructuras en XML, tales como arreglos.

- El modelo descentralizado, esto significa que puede ser procesado por varios intermediarios.
- Características específicas para operaciones clásicas de RPC con parámetros in/out, etc.

OBJETIVOS PRIMORDIALES DE SOAP (Gomez, 2010)

- Establecer un protocolo estándar de invocación de servicios remotos, basado en protocolos estándares de Internet: HTTP (Protocolo de transporte de Hipertexto) para la transmisión y XML (lenguaje de marcado extensible) para la codificación de datos.
- Independencia de plataforma, lenguaje de desarrollo e implementación (modelo de objetos).

VENTAJAS DE SOAP (autores, 2009)

- Es sencillo de implementar, probar y usar.
- Atraviesa “firewalls” y enrutadores, pues estos “piensan” que es una comunicación HTTP.
- Tanto los datos como las funciones se describen en XML, lo que permite que el protocolo no sólo sea más fácil de utilizar sino que también sea muy sólido.
- Es independiente del sistema operativo y procesador.
- Se puede utilizar tanto de forma anónima como con autenticación (nombre/clave).
- Facilidad para utilizar cualquier lenguaje.
- No se encuentra fuertemente asociado a ningún protocolo de transporte.
- No está atado a ninguna infraestructura de objeto distribuido.
- Aprovecha los estándares existentes en la industria.
- Permite la interoperabilidad entre múltiples entornos.

DESVENTAJAS DE SOAP (autores, 2009):

- Dificultad para entender las especificaciones del protocolo, puesto que muy complejo.
- Problemas de interoperabilidad.

1.3 Lenguaje UML.

Los modelos se utilizan en muchas actividades de la vida diaria ya que es una manera de abstraer una realidad compleja sobre unos bocetos, gráficos, etc. El Lenguaje Unificado de Modelado (UML, Unified Modeling Language) es un lenguaje grafico para visualizar, especificar, construir y documentar los artefactos de un sistema, cubriendo tanto las cosas conceptuales, tales como procesos del negocio y funciones del sistema, como las cosas concretas, tales como las clases escritas en un lenguaje de programación específico, esquema de bases de datos y componentes software reutilizable.

UML tiene distintos puntos de vistas de la realidad que modela mediante los distintos tipos de diagramas que posee, además permite abstraer cualquier tipo de sistema, sea informático o no, mediante los diagramas, que contienen toda la información relevante del sistema ilustrada a través de objetos, y las relaciones entre estos. Fue desarrollada en conjunto por Grady Booch, Ivar Jacobson y Jim Rumbaugh en la Rational Software Corporation, con contribuciones de otros excelentes metodólogos, vendedores de software y muchos usuarios. Basado en el amplio uso de Booch, OMT y métodos de Jacobson, el UML es la evaluación de estos y otros métodos para el modelado de procesos de negocios, objetos y componentes.

Brinda el lenguaje de aplicación de modelado para:

- Modelado de procesos de negocio con casos de uso.
- Modelado de clases y objetos.
- Modelado de componentes.
- Modelado de distribución y despliegue.

El UML estándar está compuesto por tres partes: bloques de construcción (tales como clases, objetos, mensaje), relaciones entre los bloques (tales como asociación, generalización) y diagramas (por ejemplo diagrama de actividad). Presenta nueve diagramas estándares: diagrama de casos de uso, de clases, de secuencia, de colaboración, de actividad, de estados, de implementación (componentes), de despliegue y el diagrama de objetos.

UML es el lenguaje estándar con el que es posible modelar todos los componentes del proceso de desarrollo de aplicaciones. Sin embargo hay que tener en cuenta que el modelo no pretende definir un modelo estándar de desarrollo, sino únicamente un lenguaje de modelado. Otros métodos de modelaje como OMT (ObjectModelingTechnique) o Booch si definen métodos concretos. En UML los procesos de desarrollo son diferentes según los distintos dominios de trabajo.

No define un proceso concreto que determine la fases de desarrollo de un sistema, las empresas pueden utilizar UML como el lenguaje para definir sus propios procesos y lo único que tendrán en común con otras organizaciones serán los tipos de diagrama, pues es un método independiente del proceso. (Zapata, 2009)

Capítulo 2. “Propuesta de Solución”.

Introducción.

En este capítulo se hace el análisis del sistema construir a partir del funcionamiento del negocio actual y de los requerimientos de los usuarios. Después de identificadas las necesidades se analizan las funcionalidades que debe tener el sistema que forman parte de la propuesta de solución además se reflejan los aspectos fundamentales de la solución al problema presentado.

2.1 Diagrama AS-IS

Los modelos de negocio se pueden representar a través de modelos AS-IS (Tal como es) de la organización.

Ya no se piensa que los procesos puedan diseñarse con una estructura ideal, que permanezcan inmutables con el paso de los años. Por el contrario, estos procesos están constantemente sometidos a revisiones, ya que como todo proceso, pueden ser mejorables.

El modelo As-Is además ayuda a generar un entendimiento entre las distintas áreas y locaciones de la empresa en cuanto a cómo efectivamente se ejecuta el proceso de negocios. A menudo en las organizaciones grandes muchos ejecutivos y usuarios claves no tienen la visión completa de cada uno de los pasos y detalles de la operación del proceso de negocios. La documentación del As-Is ayuda a generar claridad respecto a cómo se ejecutan las cosas y cuáles son los des alineamientos. Permite establecer los puntos críticos y de mejoramiento del proceso.



Fig. 2.1 Diagrama As-IS.

2.2 Diagrama To-Be.

La generación de los modelos To-Be es indispensable para establecer que se quiere de la nueva implementación, y ayuda a definir el nuevo modelo del proceso de negocios independientemente del software a utilizar.

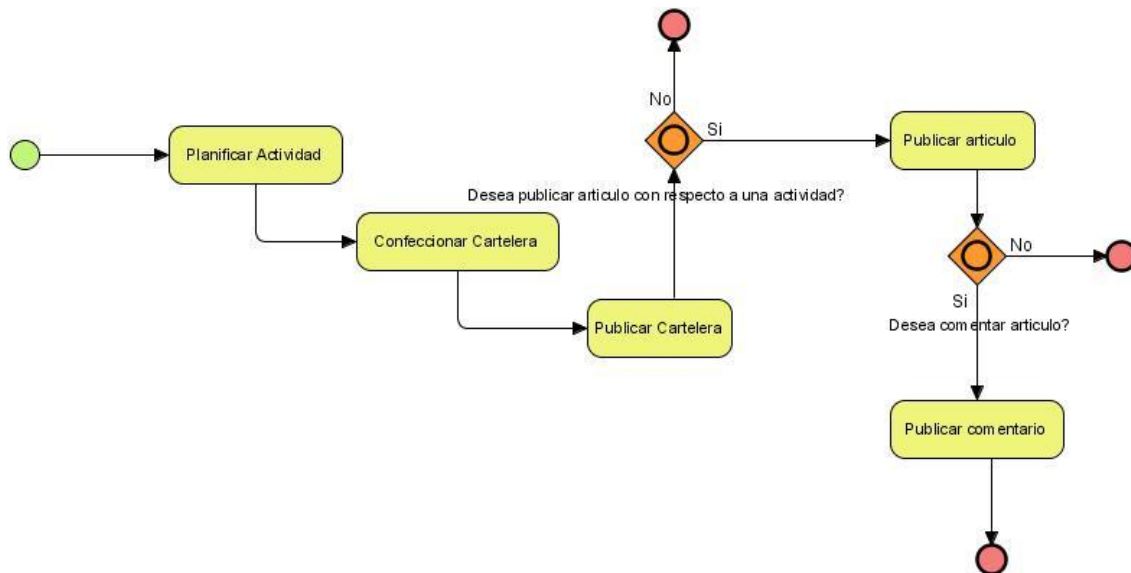


Fig. 2.2 Diagrama To-Be.

2.3 Requisitos del sistema.

2.3.1 Requisitos funcionales.

Un requisito funcional define el comportamiento interno del software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica. Son complementados por los requisitos no funcionales, que se enfocan en cambio en el diseño o la implementación.

R1- Consultar cartelera del Patio.

R2- Consultar cartelera de la sala de teatro, café y galería de arte.

R2- Leer artículos publicados.

R3- Ver imágenes de la Galería de Imágenes.

R4- Buscar artículos publicados anteriormente.

R5- Publicar comentario.

R6- Gestionar artículos.

R6.1- Eliminar artículo.

R6.2- Editar artículo

R6.3- Adicionar artículo

R7- Gestionar imágenes.

R7.1- Adicionar imagen

R7.2- Eliminar imagen.

R8- Gestionar usuarios.

R8.1- Agregar usuarios.

R8.2- Eliminar usuarios

R8.3- Editar usuarios.

R8.4- Activar usuarios.

R8.5- Buscar usuario activado y sin activar.

R9- Gestionar actividades a la cartelera.

R9.1- Agregar actividad.

R9.2- Eliminar actividad.

R9.3- Editar actividad.

2.3.2 Requisitos no funcionales.

Los requisitos no funcionales especifican las propiedades del sistema que tienen que ver con rendimiento, velocidad, uso de memoria, plataforma y fiabilidad; son las propiedades o cualidades que un producto de software debe poseer (CEIS., 2007). La mayoría de las veces se encuentran vinculados son los requisitos funcionales, ya que después que se determina qué funcionalidades que debe brindar el sistema, se especifica cómo debe comportarse, su grado de confiabilidad, su apariencia ante los clientes a los que brindará servicio, su velocidad de trabajo, la seguridad de la información que gestiona y las cualidades que poseerá.

Los requisitos no funcionales no alteran el desarrollo del producto de software, ya que no representan actividades fundamentales para el mismo, y su existencia se debe fundamentalmente a determinadas exigencias del cliente en cuanto a cualidades y manera de funcionar el sistema.

Los requerimientos no funcionales sirven de apoyo a los usuarios del sistema para valorar el mismo, ya que un producto seguro, usable, agradable y conveniente será más visitado y empleado que otro que carezca de estas cualidades.

Los requisitos no funcionales del sistema propuesto son:

Apariencia o interfaz externa: el sistema debe contar con una interfaz amigable, o sea, asequible a los usuarios ya que será un sistema que será publicado para todo tipo de usuario incluso fuera de las fronteras o sea que el diseño de la interfaz debe ser claro y expresar con imágenes la mayoría de las funcionalidades. Debe poseer un diseño respetable, de acuerdo al trabajo con el que se corresponde. Esta interfaz debe responder al cumplimiento de los requisitos planteados por los usuarios, o sea, que le debe permitir a los mismos entrar y recibir los datos de forma legible, fácil e interactiva.

Usabilidad: Con la introducción de este nuevo sistema el centro garantizará una mejor divulgación de la información, por lo que el sistema debe contar con un elevado nivel de aceptación por parte de los usuarios. El sistema se encuentra confeccionado de forma tal que puede ser utilizado por usuarios con bajo nivel de conocimientos informáticos, por lo que esto no constituye una limitante para la utilización del mismo.

Rendimiento: La mayoría de las funcionalidades que brinda el sistema necesitan de una respuesta instantánea, por lo que se garantizará la no ocurrencia de demoras. La información deberá estar disponible las 24 horas del día y se mostrará de forma clara y precisa. Una caída momentánea del sistema no provocará error en sus funcionalidades; el sistema deberá recuperarse en un corto período de tiempo ante una falla.

Soporte: El sistema contará con una base de datos donde se encontrarán los usuarios del sistema, los artículos publicados, los comentarios sobre los artículos ya mencionados, las actividades programadas para una semana y además con una aplicación Web la cual se servirá de dicha base de datos y de otra a través de un Servicio Web para compartir las carteleras y sean publicadas en otros sitios web.

La base de datos y la aplicación Web pueden encontrarse en el mismo servidor.

Portabilidad: La portabilidad no es un problema para este tipo de sistemas, ya que no necesitan ser instalados, solo necesitan un navegador web para que sea usado por los usuarios.

Seguridad: Se garantiza un control riguroso de la seguridad de la información de acuerdo a los niveles de acceso del sistema, ya que no se permitirán publicaciones sin autorización al sistema, para ello se define una política de usuarios con roles y privilegios diferentes, garantizando que la información sea consultada por las personas autorizadas. Para ellos se definieron un grupo de actores que serán descritos más adelante.

Confiabilidad: El sistema debe brindarles a los usuarios un 100% de confiabilidad sobre los datos que recibe y que se le introduce a la aplicación.

Ayuda y documentación: La aplicación desarrollada se ha implementado bastante clara y sencilla y cuenta con una interfaz lo suficientemente amigable para que el usuario, por muy pocos conocimientos informáticos que posea, se encuentre a gusto a la hora de navegar por el sitio, por lo que no se hizo necesaria la implementación de una ayuda muy detallada para el sistema.

Requerimientos Legales: Este sistema posee total legalidad y aprobación por el personal del centro, puesto que no constituye ninguna violación a leyes y la información es correctamente manejada y válida. El sistema fue confeccionado con software libre por lo que no tiene problemas con las licencias de los software utilizados en su confección, los cuales nuestro país no puede adquirir producto del bloqueo.

Requerimientos de Software:

- **Cliente:** Navegador Internet Explore versiones 8 o superior, Netscape Navegater, Moxilla Firefox, Google Chrome, Opera y todos en sus versiones 4.0 o superior.
- **Servidor:** Sistema de gestión de Bases de Datos MySQL, servidor de aplicaciones Apache que soporte tecnología PHP4 y PHP5.

Requerimientos de hardware:

- **Cliente:** como mínimo un Pentium, con 256 MB de memoria RAM y 100 MHz de velocidad del procesador.
- **Servidor:** como mínimo un Pentium III, con 512 MB de RAM y 300 MHz de velocidad del procesador.

2.4 Actores del sistema a automatizar.

A continuación se definen los actores del sistema que serán todos los usuarios que van a tener un rol en el sitio web.

ACTOR	DEFINICION
Usuario anónimo	Se considera a las personas que visiten el sitio sin autenticarse. Estos usuarios tienen disponibles los artículos publicados, así como las imágenes vinculadas al mismo, puede consultar la cartelera y efectuar búsquedas artículos anteriormente publicados. Si un usuario de este tipo desea formar parte del sistema puede registrarse.
Usuario registrado	Se considera a las personas que ya estén registradas en el sitio. Estos usuarios tienen disponibles los artículos publicados, así como las imágenes vinculadas al mismo, comentar dichos artículos, efectuar búsquedas artículos anteriormente publicados y puede consultar la cartelera.
Publicador	Se considera a las personas que ya estén registradas en el sitio y que tengan los permisos de periodista. Estos usuarios tienen disponibles los artículos publicados, así como las imágenes vinculadas al mismo, comentar dichos artículos, efectuar búsquedas artículos anteriormente publicados y puede consultar la cartelera. Además se encargan de gestionar la publicación edición o eliminación de los artículos y cargar las imágenes vinculadas al mismo.
Administrador	El administrador es quien tiene mayor número de funcionalidades dentro del sistema, se encarga de todo lo referente a los usuarios insertar, modificar, eliminar o activar un usuario. La gestión de las actividades, para conformar la cartelera, que se planifiquen para la semana (eliminar, editar y agregar). Igualmente se encarga de gestionar la publicación, edición o eliminación de los

artículos y cargar las imágenes vinculadas al mismo. Además tiene disponibles los artículos publicados, así como las imágenes vinculadas al mismo, comentar dichos artículos, efectuar búsquedas artículos anteriormente publicados y puede consultar la cartelera.

2.5 Diagrama de casos de uso del sistema.

Los diagramas de casos de uso documentan el comportamiento del sistema desde el punto de vista de un usuario.

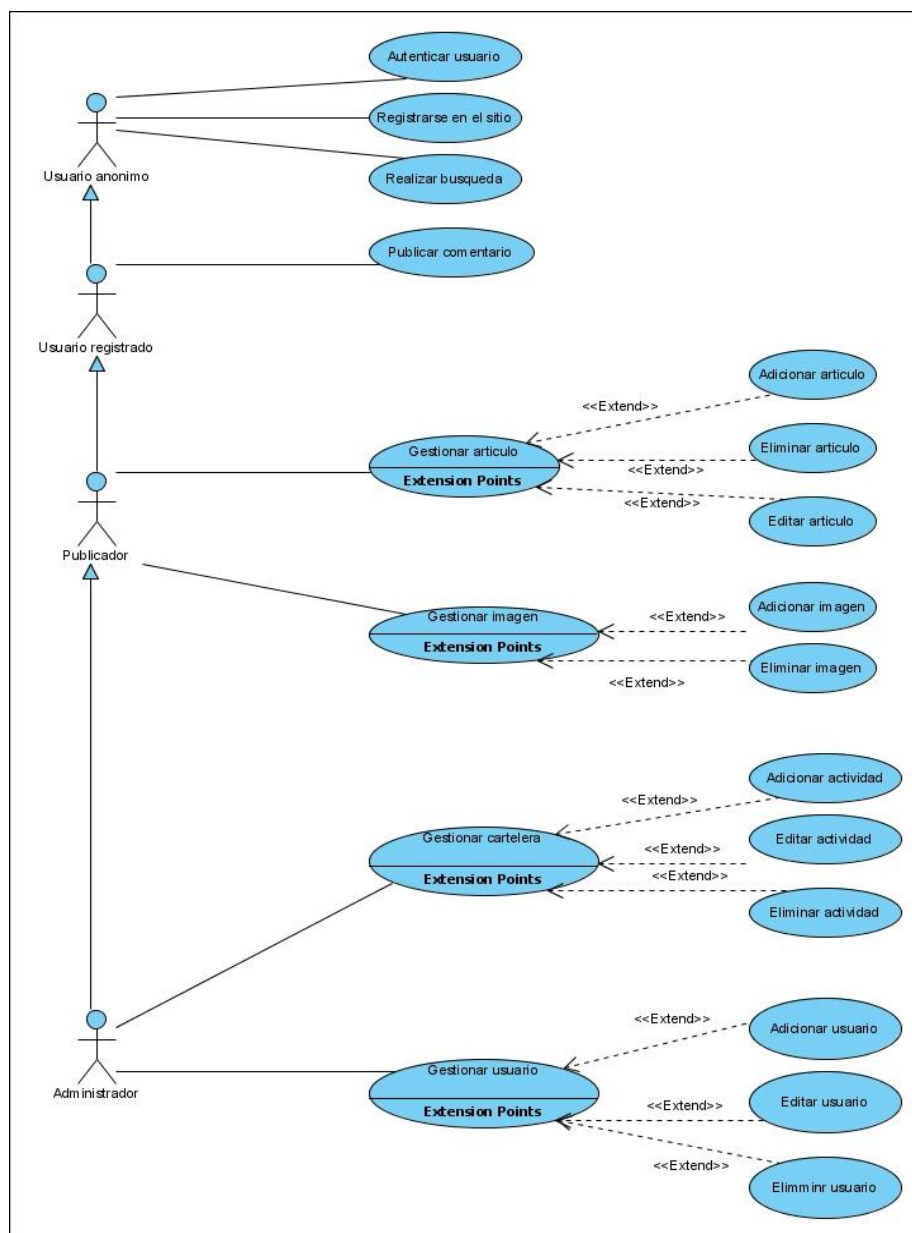


Fig. 2.3 Diagrama de casos de uso del sistema.

2.6 Nombre y descripción de los casos de uso del sistema.

A continuación se hace un análisis de todos los casos de uso del sistema, mostrando el actor que interactúa con el caso de uso, el propósito del caso de uso y un resumen detallado de del caso de uso que se está analizando.

Tabla 2.1 Descripción del caso de uso: Registrar usuario

Caso de Uso	Registrar usuario.
Actor	Usuario anónimo, usuario registrado, publicador, administrador.
Propósito	Registrarse en el sitio
Resumen	El caso de uso comienza cuando el usuario desea crear una cuenta, introduciendo sus datos en el formulario correspondiente. Luego el administrador asignará a qué tipo de usuario pertenece y habilitará la cuenta.

Tabla 2.2 Descripción del caso de uso: Realizar búsqueda.

Caso de Uso	Realizar búsqueda
Actor	Usuario Anónimo, Usuario registrado, Publicador, Administrador
Propósito	Realizar búsquedas.
Resumen	El caso de uso comienza cuando cualquiera de los actores desea acceder a los artículos que han sido publicados anteriormente, a través de una búsqueda avanzada. Si el artículo obtenido tiene imágenes relacionadas al mismo, puede acceder a ellas.

Tabla 2.3 Descripción del caso de uso: Gestionar artículo.

Caso de Uso	Gestionar artículo
Actor	Administrador, Publicador.
Propósito	Adicionar, modificar o eliminar un artículo.
Resumen	El caso de uso comienza cuando cualquiera de los actores permitidos desea publicar, eliminar o editar un artículo.

Tabla 2.4 Descripción del caso de uso: Gestionar imagen.

Caso de Uso	Gestionar imagen
Actor	Administrador, Publicador.
Propósito	Eliminar o adicionar una imagen.
Resumen	El caso de uso comienza cuando cualquiera de los actores permitidos desea publicar o eliminar una imagen relacionada a un artículo.

Tabla 2.5 Descripción del caso de uso: Gestionar cartelera.

Caso de Uso	Gestionar cartelera
Actor	Administrador.
Propósito	Adicionar, eliminar o editar una actividad de la cartelera.
Resumen	El caso de uso comienza cuando el administrador desea adicionar, eliminar o editar una actividad que conforma la cartelera.

Tabla 2.6 Descripción del caso de uso: Gestionar usuario.

Caso de Uso	Gestionar usuario
Actor	Administrador.
Propósito	Adicionar, eliminar, editar o activar un usuario.
Resumen	El caso de uso comienza cuando el administrador desea adicionar, eliminar, editar o activar un usuario.

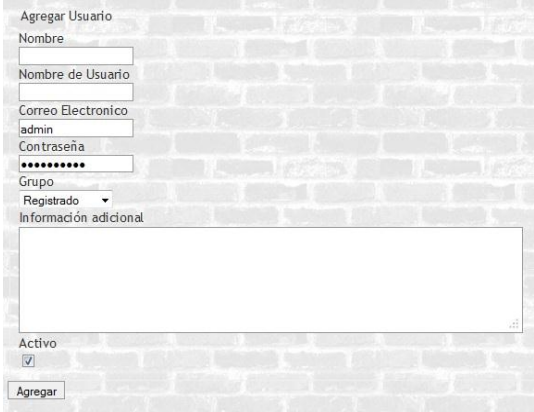
Tabla 2.7 Descripción del caso de uso: Publicar comentario.

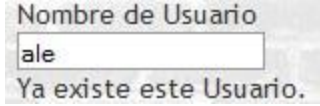
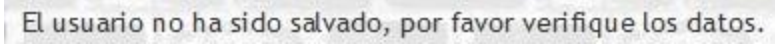
Caso de Uso	Publicar comentario
Actor	Usuario registrado, Publicador, Administrador
Propósito	Publicar, editar o eliminar comentario.
Resumen	El caso de uso comienza cuando cualquiera de los actores con los permisos pertinentes desea comentar sobre alguno de los artículos publicados.

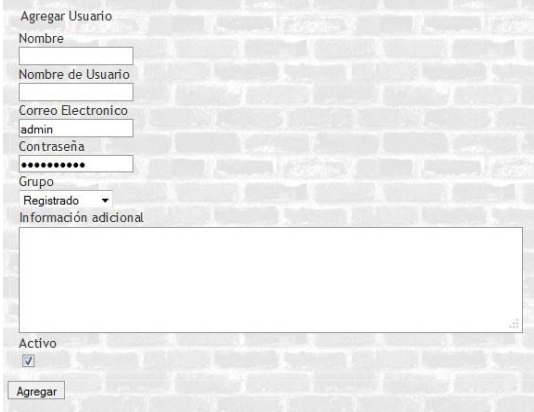
A continuación se van a analizar la descripción de alto nivel de los casos de uso gestionar usuario que es una tarea llevada a cabo por el administrador del sistema y publicar artículo que puede ser gestionada por el administrador o por el usuario publicador.

Tabla 2.8 Descripción del caso de uso: Gestionar usuario.

Caso de uso:	Gestionar usuario.
Actores:	Administrador.
Resumen:	El administrador necesita adicionar, eliminar, editar y activar usuarios.
Referencias:	R8

Prioridad:	Critica.
Precondiciones:	-El usuario tiene que estar autenticado como administrador del sistema y haber accedido a la opción “gestionar usuario”.
Flujo normal de eventos.	
Sección “adicionar usuario”.	
Acción del actor.	Respuesta del negocio.
1-El administrador accede a la opción “nuevo usuario”.	2-El sistema muestra los campos necesarios que debe llenar el administrador.
3-El administrador introduce los datos necesarios para este formulario.	4-El sistema verifica que no exista ningún campo obligatorio vacío.
	5-El sistema verifica que no exista el usuario que se está agregando.
	6- El sistema muestra un mensaje informando que se agregó el usuario correctamente.
Vista de la interfaz: usuario nuevo.	
	

Flujos Alternos.	
Acción del actor.	Respuesta del negocio.
	4.1-El sistema detecta un campo obligatorio vacío. Muestra al usuario un mensaje de error.
	5.1-El sistema detecta que el usuario ya existe. Le notifica al administrador a través de un mensaje de error.
Vista de la Interfaz: Mensaje de error.  	
Sección “modificar usuario”.	
Acción del actor.	Respuesta del negocio.
1-El administrador selecciona de la tabla el usuario que desea modificar y entra a la opción editar.	2. El sistema muestra la interfaz con todos los datos del usuario que desea modificar. Los campos se habilitan para que el usuario realice los cambios deseados.
3-El usuario realiza los cambios deseados.	4-El sistema verifica que no exista ningún campo obligatorio vacío.
	5- El sistema verifica que no exista el usuario.
	6- El sistema muestra un mensaje informando que se realizaron los cambios correctamente.
Vista de Interfaz: Usuario “Editar”	

	
Flujos alternos.	
Acción del actor.	Respuesta del negocio.
	5.1 El sistema detecta un campo obligatorio vacío. Muestra al usuario un mensaje de error.
	6.1 El sistema detecta que el usuario ya existe. Le notifica al usuario a través de un mensaje de error.
Sección “Eliminar usuario”	
Acción del actor.	Respuesta del negocio.
1- El administrador selecciona el usuario que desea eliminar	
2-El usuario accede a la opción “Borrar”.	3-El sistema muestra mensaje de confirmación de eliminación.
4-El usuario acepta la el mensaje de confirmación.	5- El sistema elimina el usuario elegido.
	6- El sistema muestra mensaje de que el usuario fue eliminado correctamente.
Vista de Interfaz: Eliminar usuario.	

Usuario	Correo	Grupo	Creado	Acciones
admin	admin@udv.edu.cu	Administrador	2012-04-16 21:06:15	Edita Eliminar
ale	ale@udv.edu.cu	Registrado	2012-06-07 22:10:29	Editar Eliminar
perio	perio@udv.edu.cu	Periodista	2012-06-07 22:43:17	Edita Eliminar
jose	admin@uuu	Registrado	2012-06-08 12:09:51	Editar Eliminar

Página 1 de 1, muestra 4 datos de 4 en total, comenzando del dato 1, terminando en 4

<< Anterior | | Siguiente >>

Acciones

- [Activar Usuarios](#)
- [Nuevo Usuario](#)

Sección “activar usuario”.

Acción del actor.

Respuesta del negocio.

1-El administrador accede a la opción “Activar usuario”.

2- El sistema muestra una tabla con los usuarios sin activar.

3- El administrador activa los usuarios.

4- El sistema muestra mensaje de que los usuarios han sido activados.

5- El sistema refresca la tabla de usuarios y muestra todos los usuarios activados.

Vista de Interfaz: Activar usuario.

Usuarios					
Usuario	Correo	Grupo	Creado	Modificado	Acciones
ya	ya@udv.edu.cu	Registrado	2012-06-11 17:14:25	2012-06-11 17:14:25	Activar

Página 1 de 1, muestra 1 datos de 1 en total, comenzando del dato 1, terminando en 1

<< Anterior | | Siguiente >>

Acciones

- [Nuevo Usuario](#)

Tabla 2.9 Descripción del caso de uso: Publicar Artículo.

Caso de uso:	Publicar Artículo.
Actores:	Administrador, Publicador.
Resumen:	El administrador o el publicador necesitan adicionar o editar.
Referencias:	R6
Prioridad:	Crítica.
Precondiciones:	-El usuario tiene que estar autenticado como administrador o publicador del sistema y haber accedido a la opción “agregar artículo”.
Flujo normal de eventos.	
Sección “adicionar artículo”.	
Acción del actor.	Respuesta del negocio.
1-El administrador o el publicador acceden a la opción “agregar artículo”.	2-El sistema muestra los campos necesarios que debe llenar el administrador o el publicador.
3-El administrador o el publicador introducen los datos necesarios para este formulario.	4-El sistema verifica que no exista ningún campo obligatorio vacío.
	5- El sistema muestra un mensaje informando que se agregó el artículo correctamente.
Vista de la interfaz: “agregar artículo”	

Flujos Alternos.

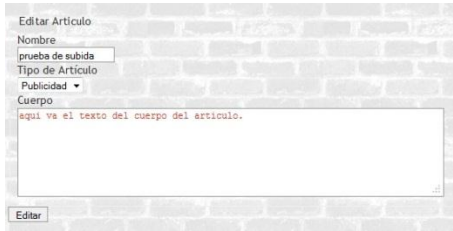
Acción del actor.	Respuesta del negocio.
	4.1-El sistema detecta un campo obligatorio vacío. Muestra al usuario un mensaje de error.
	5.1-El sistema detecta que el usuario ya existe. Le notifica al administrador a través de un mensaje de error.

Vista de la Interfaz: Mensaje de error.

El artículo no ha podido ser salvado. Por favor, pruebe de nuevo.

Sección “Editar comentario”.

Acción del actor.	Respuesta del negocio.
1-El administrador o el publicador acceden al artículo a editar.	2- El sistema muestra el formulario para editar el artículo.
3-El administrador o el publicador introducen los datos necesarios para este formulario.	4-El sistema verifica que no exista ningún campo obligatorio vacío.
	5- El sistema muestra un mensaje informando que se editó el artículo

	correctamente.
<p>Vista de la interfaz: “editar artículo”</p> 	

2.7 Diagrama de clases del análisis.

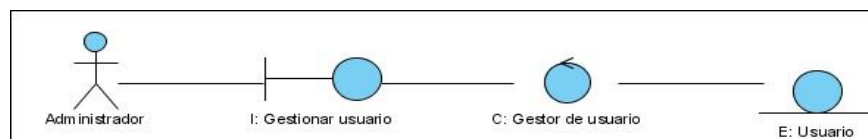


Fig. 2.4 Diagrama de clases del análisis del caso de uso Gestionar Usuario.

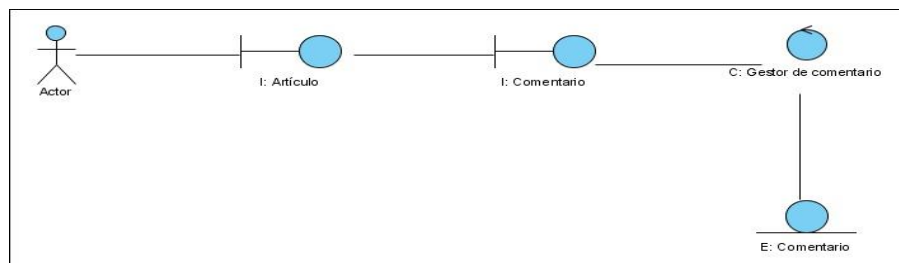


Fig. 2.5 Diagrama de clases del análisis del caso de uso Publicar comentario.

2.8 Diagrama de colaboración.

El diagrama describe qué es lo que tiene lugar en los objetos participantes en términos de cómo los objetos se comunican enviándose mensajes unos a otros. Se puede hacer un diagrama de colaboración para cada variante de flujo de eventos de un caso de uso.

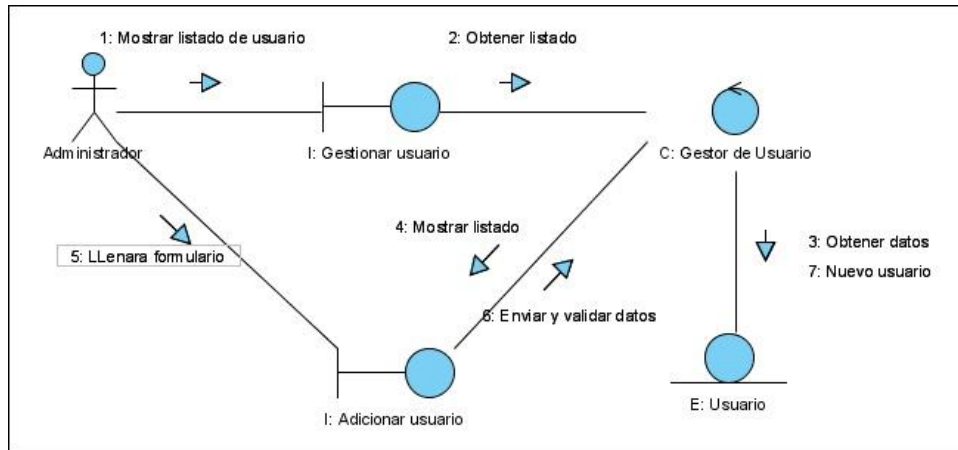


Fig. 2.6 Diagrama de colaboración del caso de uso Agregar usuario.

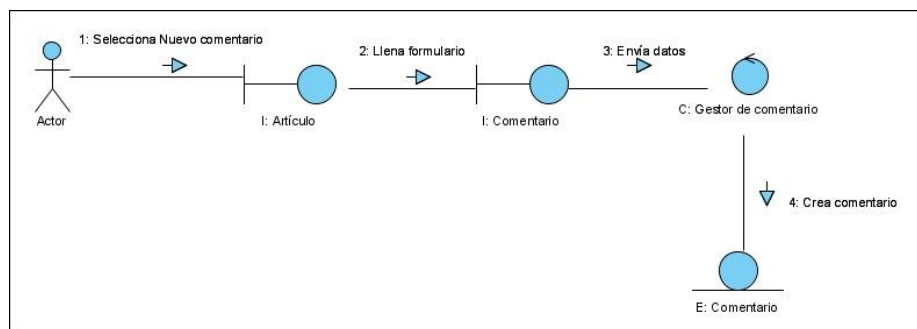


Fig. 2.7 Diagrama de colaboración del caso de uso Agregar comentario.

2.9 Diagrama de secuencia.

Los diagramas de secuencia son particularmente importantes para los diseñadores debido a que ellos aclaran los roles de los objetos en el flujo y por consiguiente brindan la entrada básica para la determinación de las responsabilidades y las interfaces de las clases.

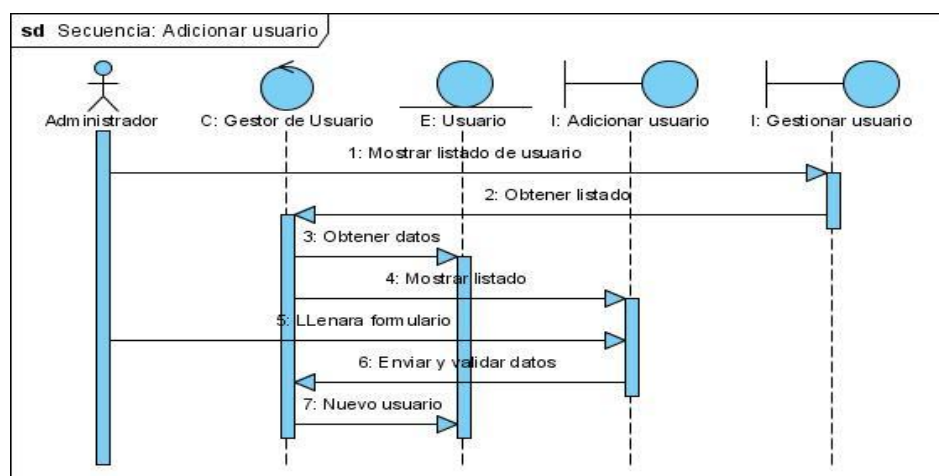


Fig. 2.8 Diagrama de secuencia adicionar usuario.

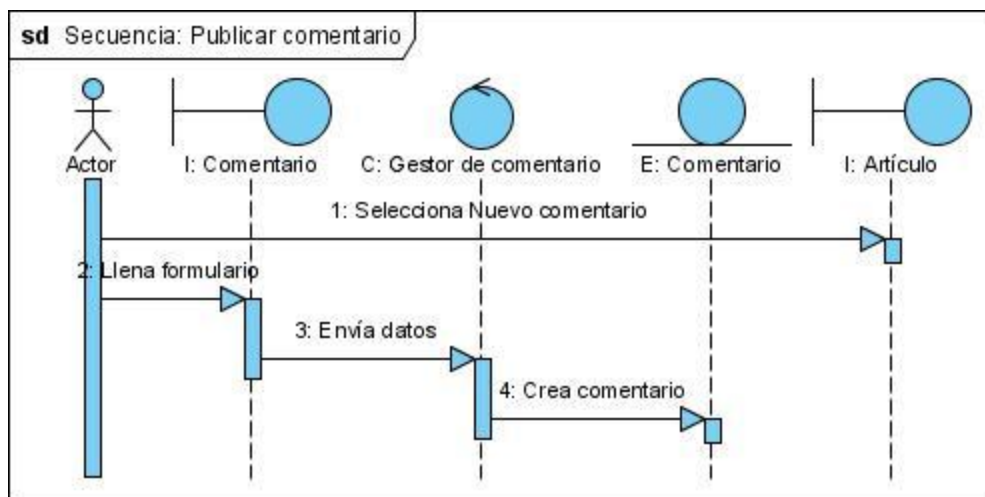


Fig. 2.9 Diagrama de secuencia publicar comentario.

2.10 Diagrama Entidad-Relación.

Es una técnica de análisis (modelado) de arriba-abajo basada en la identificación de las entidades y de las relaciones que se dan entre ellas en la parte de realidad que pretendemos modelar. Describe el “mundo real” como un conjunto de ENTIDADES y de RELACIONES entre ellas.

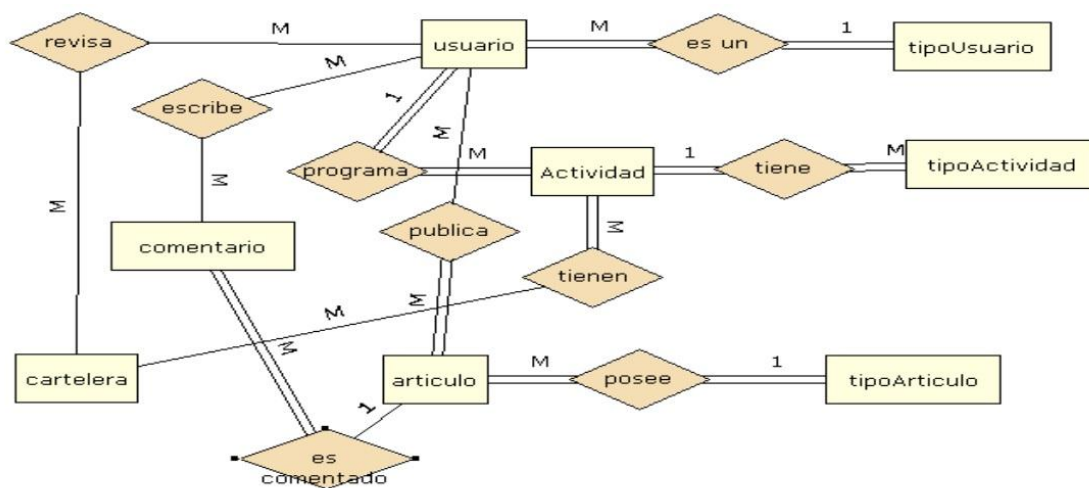


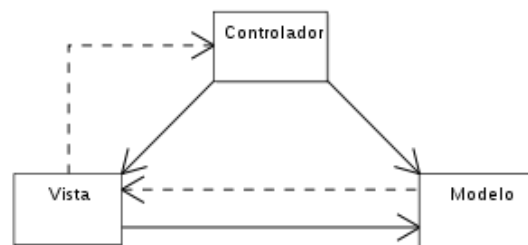
Fig. 2.10 Modelo de entidad relación de la Base de datos.

2.11 Arquitectura M-V-C.

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres

componentes distintos. El patrón de llamada y retorno MVC (según CMU), se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.(wikipedia, 2011a)

Fue diseñada para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Este modelo de arquitectura se puede emplear en sistemas de representación gráfica de datos, o en sistemas CAD, en donde se presentan partes del diseño con diferente escala de aumento, en ventanas separadas. En la figura siguiente, vemos la arquitectura MVC en su forma más general.(Catalani, 2008)



Ventajas.

Este modelo de arquitectura presenta varias ventajas:

- Hay una clara separación entre los componentes de un programa; lo cual nos permite implementarlos por separado
- Hay un API muy bien definido; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

Descripción del patrón

Modelo: Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la *vista* y su *controlador* facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.

En el sistema cada modelo representa cada una de las tablas de la base de datos. En el modelo se encuentra la estructura de las tablas, es decir cada uno de los campos que tiene la tabla y están todas las relaciones posibles con otras tablas. En la figura a continuación se muestra parte de un modelo.

```
'user_id' => array(
    'numeric' => array(
        'rule' => array('numeric'),
        // 'message' => 'Your custom message here',
        // 'allowEmpty' => false,
        // 'required' => false,
        // 'last' => false, // Stop validation after this rule
        // 'on' => 'create', // Limit validation to 'create' or 'update' operations
    ),
),
);
//The Associations below have been created with all possible keys, those that are not needed can be removed

var $belongsTo = array(
    'Articulo' => array(
        'className' => 'Articulo',
        'foreignKey' => 'articulo_id',
        'conditions' => '',
        'fields' => '',
        'order' => ''
    ),
    'User' => array(
        'className' => 'User',
        'foreignKey' => 'user_id',
        'conditions' => '',
        'fields' => '',
        'order' => ''
    )
);
```

Fig. 2.11 Campo identificador del usuario de la tabla comentario y asociación con las tablas artículos y usuario.

Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente con la interfaz de usuario.

En el sistema la vista es la que se encarga de hacer la llamada a las acciones que realiza el usuario y mostrarlas, están dentro de carpetas que tienen el mismo nombre del modelo que le corresponde solo que en plural y programadas con código PHP, existe un archivo vista (ctp) para cada función que pueda accionar un usuario. Seguidamente se muestra el código de la vista correspondiente a agregar comentario.

```
<div class="comentarios form">
    <?php echo $this->Form->create('Comentario');?>
    <fieldset>
        <legend><?php __('Agregar Comentario'); ?></legend>
        <?php
            echo $this->Form->input('body', array('label' => 'Su comentario'));
            echo $this->Form->hidden('articulo_id', array('value' => $articulo_id));
        ?>
    </fieldset>
    <?php echo $this->Form->end(__('Agregar', true));?>
</div>
```

Fig. 2.12 Código de la vista agregar comentario.

Controlador: Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.

En el sistema un controlador es el que posee toda la lógica de la aplicación y comunica las vistas con los modelos. Hay un archivo controlador por cada modelo que exista, a continuación se muestra el código del controlador agregar comentario.

```
function add( $articulo_id = NULL ) {
    if ( !empty($this->data) ) {
        $this->Comentario->create();
        $this->data['Comentario']['user_id'] = $this->Session->read('Auth.User.id');
        if ($this->Comentario->save($this->data)) {
            $this->Session->setFlash(__( 'El comentario ha sido salvado', true ));
            $this->redirect(array( 'controller' => 'articulos', 'action' => 'view' , $this->data['Comentario']['articulo_id']));
        } else {
            $this->Session->setFlash(__( 'El comentario no ha podido ser salvado, por favor pruebe de nuevo.', true ));
        }
    }
    $this->set(compact( /*'articulos', 'users', */ '/'articulo_id' ));
}
```

Fig. 2.13 Código agregar comentario del controlador

Muchos de los sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos: en líneas generales del **MVC** corresponde al modelo. La unión entre *capa de presentación* y *capa de negocio* conocido en el paradigma de la Programación por capas representaría la integración entre **Vista** y su correspondiente **Controlador** de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la *capa visual gráfica* de su correspondiente *programación y acceso a datos*, algo que mejora el desarrollo y mantenimiento de la *Vista* y el *Controlador* en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

Aunque se pueden encontrar diferentes implementaciones de **MVC**, el flujo que sigue el control generalmente es el siguiente:

1. El usuario interactúa con la interfaz de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador

actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.

4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón Observador para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. En general el controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

Marcos de trabajos que utilizan M-V-C.

Lenguaje	Marco de trabajo
PHP	CakePHP
PHP	Symfony
PYTHON	Django
.NET	Spring .NET
PHP	Zend Framework
JAVA	Aurora
JAVA	Spring

2.12 Modelo de componente.

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.

Debido a que los diagramas de componentes son más parecidos a los diagramas de casos de usos, éstos son utilizados para modelar la vista estática y dinámica de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

En él se situarán librerías, tablas, archivos, ejecutables y documentos que formen parte del sistema.

Uno de los usos principales es que puede servir para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

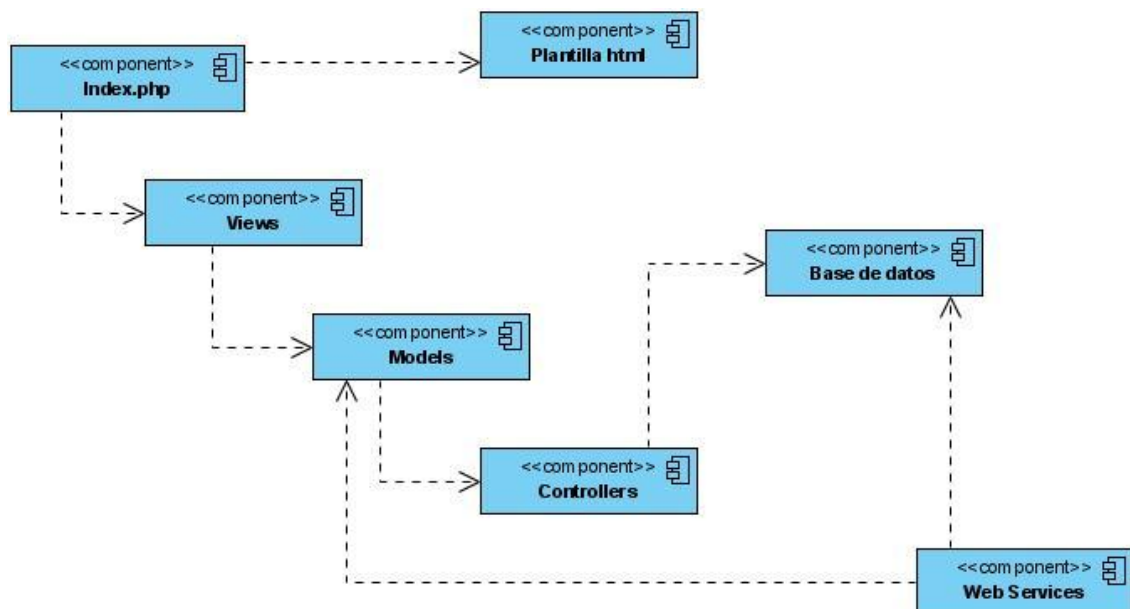


Fig. 2.14 Diagrama del modelo de componente.

2.13 Modelo de despliegue.

El modelo de despliegue define la arquitectura física del sistema por medio de nodos interconectados. Estos nodos son elementos hardware sobre los cuales pueden ejecutarse los elementos software. Con frecuencia conocemos cómo será la arquitectura física del sistema antes de comenzar su desarrollo. Por tanto, podemos modelar los nodos y las conexiones del modelo de despliegue tan pronto como comience el flujo de trabajo de los requisitos.

Durante el diseño, decidiremos qué clases son activas, es decir, son hilos o procesos. Determinaremos lo que debería hacer cada clase activa, cómo debería ser el ciclo de vida de cada una de ellas, y cómo deberían comunicarse, sincronizarse, y compartir información. Los objetos activos se asignan a los nodos del modelo de despliegue. Al hacer esta asignación, debemos considerar la capacidad de los nodos, como su capacidad de proceso y tamaño de memoria, y las características de las conexiones, como el ancho de banda y la disponibilidad.

Los nodos y conexiones del modelo de despliegue y la asignación de los objetos activos a los nodos pueden mostrarse en diagramas de despliegue. Estos diagramas también pueden mostrar cómo se asignan los componentes ejecutables a los nodos [19].



Fig. 2.15 Diagrama de despliegue.

Capítulo 3. “Manual del usuario”

El resultado de esta investigación es una aplicación Web como se ha venido insistiendo desde capítulos a anteriores. Una de las características principales de este tipo de sistemas es que su diseño de interfaz sea amigable, y sugerente de acorde a sus funcionalidades para cualquier tipo de usuario que pueda acceder a él.

A continuación se muestran las pantallas principales del sistema en dependencia de las funcionalidades que brindan.



Fig. 3.1 Página de Inicio del Sitio



Fig. 3.2 Cartelera cultural de la semana del patio.



Fig. 3.3 Interfaz de un artículo y sus comentarios.

3.1 Manual de usuario para los servicios web.

El sitio web implementado posee un servicio web para compartir la cartelera que se publique y que otras entidades puedan acceder y propagarlo en sus espacios en la red. En la siguiente figura se muestra la parte inferior donde se encuentra el enlace para obtener el cliente y la petición de ya hablado servicio.

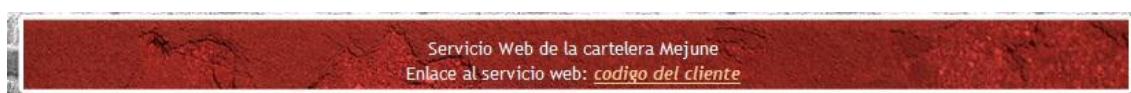


Fig. 3.4 Enlace al servicio web.

El servicio web fue creado para que paginas como la intranet de la UCLV, el periódico vanguardia digital, entre otros puedan incluir en su sitio la cartelera del mejunje, de esta forma la divulgación de las actividades que se realizan en el centro llegaría no solo por la web del centro si no por cualquier página que posea este servicio.

3.1.1 Procedimiento para consumirlo.

El sitio de “El Mejunje de Silverio” tiene en la parte inferior un vínculo a dos archivos que contienen el código para hacer la petición para compartir la cartelera como uno de sus servicios en su web. El usuario interesado en comunicar dicha información tiene que crear un archivo de extensión *.wsdl* que contendrá el código que hace la petición del servicio web de la cartelera. Otro archivo a crear tendría extensión *.php* que sería el cliente el cual contendría lo que desea mostrar en su servicio.

Al acceder a la opción “*código del cliente*” se muestra el código que hará la petición al sitio de “El Mejunje de Silverio” para compartir la cartelera semanal. La estructura de esta prestación se muestra en forma de tabla la cual puede ser modificada a gusto del consumidor.

NOMBRE	DESCRIPCION	FECHA	HORA
actividad 1	La trovuntivitis con sus invitados	2012-06-14	10:21:00
actividad 2	el Guayabazo	2012-06-14	10:21:00
actividad 3	El mejunje no cree en lagrimas	2012-06-14	10:22:00

Fig. 3.5 Interfaz del servicio web.

3.2 Evaluación de calidad web.

Pruebas de Calidad de Software, son los procesos que permiten verificar y revelar la calidad de un producto. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un programa. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas.

Las pruebas de calidad de software se integran en las diferentes fases del ciclo del software dentro de la Ingeniería de software. Así se ejecuta un programa y mediante técnicas experimentales se trata de descubrir que errores tiene, además pueden permitir que un software defectuoso no llegue al cliente.

3.2.1 Importancia de hacer las pruebas de calidad.

Toda prueba de software desempeña un papel fundamental en el desarrollo de cualquier tipo de aplicación. Estas pruebas son un elemento importante para la garantía de calidad del software y presenta una revisión final de las especificaciones, del diseño y de la codificación.

1. El sistema es entregado al cliente con todas las funcionalidades trabajando de manera adecuada.
2. Es una manera de comprobar que no existen errores en el sistema.
3. Se comprueba que no va tener incompatibilidades con distintos entornos de hardware o de sistemas operativos.
4. Mostrar hasta qué punto las funciones del software operan de acuerdo con las especificaciones y requisitos del cliente.

3.2.2 Tipos de Pruebas de Software

Las pruebas en conjunto tienen como objetivo general verificar y validar un software, independientemente de las características y el entorno donde se desarrollen, además de los recursos y los factores vinculados al proceso de desarrollo. (Ecured, 2009a)

1-Funcionalidad

- Pruebas fijando su atención en la validación de las funciones, métodos, servicios, caso de uso.
- Asegurar que los datos o el sistema sean solamente accedido por los actores deseados.

2-Usabilidad

Prueba enfocada a factores humanos, estéticos, consistencia en la interfaz de usuario, ayuda sensitiva al contexto y en línea, asistente documentación de usuarios y materiales de entrenamiento.

3-Fiabilidad

- Capacidad de un sistema para mantener su nivel de rendimiento.
- Valoración a la adherencia a su diseño y formación. Este tipo de prueba es hecho a las aplicaciones Web asegurando que todos los enlaces están conectados, el contenido deseado es mostrado y no hay contenido huérfano.
- Evaluar cómo el sistema responde bajo condiciones anormales. (extrema sobrecarga, insuficiente memoria, servicios y hardware no disponible, recursos compartidos no disponible).

4-Rendimiento

- Benchmark: Es un tipo de prueba que compara el rendimiento de un elemento nuevo o desconocido a uno de carga de trabajo de referencia conocido.
- Contención: Enfocada a la validación de las habilidades del elemento a probar para manejar aceptablemente la demanda de múltiples actores sobre un mismo recurso (registro de recursos, memoria).
- Carga: Usada para validar y valorar la aceptabilidad de los límites operacionales de un sistema bajo carga de trabajo variable, mientras el sistema bajo prueba permanece constante. La variación en carga es simular la carga de trabajo promedio y con picos que ocurre dentro de tolerancias operacionales normales.

5-Soportabilidad

- Configuración: Enfocada a asegurar que funciona en diferentes configuraciones de hardware y software. Esta prueba es implementada también como prueba de rendimiento del sistema.
- Instalación: Enfocada a asegurar la instalación en diferentes configuraciones de hardware y software bajo diferentes condiciones (insuficiente espacio en disco, etc.

3.3 Pruebas de calidad del Software hechas al sistema.

El sitio web fue sometido a tres pruebas de calidad de software.

Usabilidad: el sistema web fue creado para una institución que posee características especiales tanto arquitectónicamente como en los servicios que ofrece, por tanto se necesitaba construir una interfaz que estuviera acorde con estas particularidades. Tanto el diseño de interfaz del sistema así como las funcionalidades fueron sometidas criterios de 25 usuarios entre ellos solo 6 se desenvuelven en las ramas de la informática.

Todas estas personas encuestadas coinciden que a primera vista el sistema refleja el lugar al que se referencia, además no tuvieron complicaciones en navegar dentro del sistema aunque algunas de las personas sugirieron otras funcionalidades que pueden ser incorporadas.

A parte de las personas encuestadas un factor importante es el juicio del cliente quien quedo complacido con el producto presentado.

Portabilidad: este sitio web fue creado con una herramienta de software libre que permite que funcione en cualquier sistema operativo tanto en alguna de las versiones de Windows como en Linux. El servidor donde se ejecuta el sistema, en este caso Xampp, también es una herramienta primordial que permite su instalación en varios sistemas operativos. El sitio web creado fue instalado en varios sistemas operativos como son Windows XP, Windows Server 2008 y Windows 7. En cualquiera de estos sistemas operativos el sitio no perdió ninguna de sus características visuales, ni presentó problemas tanto de conexión a la base de datos o algún error a la hora de instalarlo.

Eficiencia

Para hacer las pruebas de eficiencia nos basamos en los distintos entornos de red desde los que se puede acceder al sitio. Estos entornos fueron creados de forma virtual con la herramienta Dummynet. Es originalmente una herramienta diseñada por probar los protocolos de la gestión de redes, y desde entonces usó para una variedad de aplicaciones incluso la dirección del ancho de banda. Este útil se puede usar con varios sistemas operativos como son Linux, Mac OS X y las versiones de Windows.

El dummynet trabaja interceptando los paquetes a través de uno o más objetos llamados cañerías que simulan los efectos de las limitaciones del ancho de banda que la propagación tarda y las pérdidas de paquetes. (GSP, 2002)

Pasos seguidos para llevar a cabo la prueba con el programa dummynet:

1. Instalar el software.
2. Se configura una conexión, es decir crear los entornos o puentes de red diferentes, introduciendo códigos por la consola CMD.
3. Tener la aplicación web ejecutándose en otra computadora.
4. Luego con el navegador Google Chrome (usado para hacer las pruebas) se hacen los distintos llamados a la web y este proporciona las estadísticas del tiempo que demora el acceso a la misma.
5. Se borra el historial del navegador para que no quede ninguna huella del sitio y se pueda aplicar la prueba nuevamente desde el paso número 2 cambiando la velocidad de los puentes.

Siguiendo el estándar utilizado por el buscador Google se creó un estándar para medir la eficiencia teniendo en cuenta los tipos de conexiones más comunes en el país. Por lo que se creó los siguientes intervalos para la clasificación de la eficiencia según el tiempo de espera y los entornos de conexiones posibles.

El tiempo de respuesta del servidor al cliente depende del entorno de red que esté utilizando entonces:

1. Si el tiempo que demora el sitio en mostrarse completo esta entre 0 a 5000: la eficiencia es muy buena.
2. Si el tiempo que demora el sitio en mostrarse completo esta entre 5000 a 50000: la eficiencia es buena.
3. Si el tiempo que demora el sitio en mostrarse completo esta entre 50000 a 105000: la eficiencia es regular.
4. Si el tiempo que demora el sitio en mostrarse completo es mayor que 105000: la eficiencia es mala.

A continuación se muestra una tabla con los resultados arrojados por las pruebas realizadas.

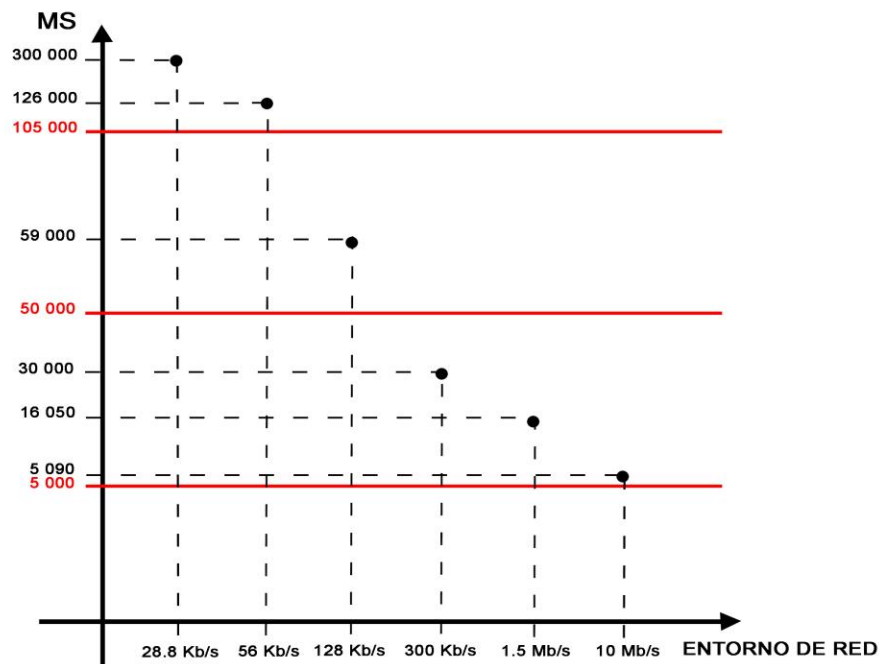


Fig. 3.6 Grafica tiempo de espera con respecto al entorno de red usado.

3.4 Estudio de factibilidad

3.4.1 Introducción.

En los capítulos anteriores se realizó un estudio de todo el proceso en general de como en la actualidad funciona en el centro cultural, así como la implementación de un sistema Web al cual tengan acceso la mayoría de las personas y entidades para así lograr una mejor comunicación entre el “Mejunje de Silverio”, el centro provincial de cultura, y todas las personas interesadas en el complejo cultural de forma más interactiva y eficiente. En el presente capítulo se llevará a cabo un profundo estudio económico de este software antes mencionado, para valorar la factibilidad del mismo y si resulta realmente económica la puesta en marcha de su funcionamiento.

3.4.2 Planificación basada en casos de uso.

La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner de Objectory AB, y posteriormente refinado por muchos otros autores. Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. (Peralta)

3.4.3.1 Cálculo de puntos de casos de uso sin ajustar (UUCP)

Para la estimación el primer paso que se lleva a cabo es el cálculo de los Puntos de Casos de Uso sin ajustar. Este valor se calcula a partir de la siguiente ecuación:

$$UUCP = UAW + UUCW \quad \text{donde,}$$

UUCP: Puntos de Casos de Uso sin ajustar

UAW: Factor de Peso de los Actores sin ajustar

UUCW: Factor de Peso de los Casos de Uso sin ajustar

3.4.3.1.1 Determinación del factor peso de los actores sin ajustar (UAW)

Este valor se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los actores se establece, teniendo en cuenta en primer lugar, si se trata de una persona o de otro sistema, y en segundo lugar, la forma en que el actor interactúa con el sistema (Peralta)

Tabla 3.1 Factores de peso de los actores.

Tipo de actor	Descripción	Factor de peso	Número de actores	Resultado
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación.	1	0	0
Promedio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0	0
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	4	12

	<i>Total</i>	<i>12</i>
--	--------------	-----------

$$UAW = 12$$

3.4.3.1.2 Determinación del factor peso en los casos de uso sin ajustar (UUCW).

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los casos de uso se establecen teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómicas, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia. (Peralta)

Tabla 3.2 Factores de peso de los casos de uso.

Tipo de caso de uso	Descripción	Factor de peso	Número de Casos de Uso	Resultado
Simple	1-3 Transacciones	5	9	45
Promedio	4-7 Transacciones	10	6	60
Complejo	Mayor de 8 Transacciones.	15	0	0
			<i>Total</i>	<i>105</i>

$$UUCW = 105$$

Calculando

$$UUCP = UAW + UUCW$$

$$UUCP = 12 + 105$$

$$UUCP = 117$$

3.4.3 Cálculo de puntos de casos de uso ajustados.

Seguidamente de calcular los Puntos de Casos de Uso sin ajustar, se debe ajustar este valor mediante la siguiente ecuación:

$$UCP = UUCP \times TCF \times EF \text{ donde,}$$

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

3.4.3.1 Determinación del factor de complejidad técnica (TCF)

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

Tabla 3.3. Factores de complejidad técnica.

Número de factor	Descripción	Peso	Valor	Factor	Comentario
T1	Sistema Distribuido	2	1	2	<i>El sistema es Web, por lo que posee cierto nivel de distribución</i>
T2	Tiempo de respuesta	1	1	1	<i>El tiempo de respuesta respalda los objetivos que se persiguen con el proyecto realizado, por lo que es el adecuado.</i>
T3	Eficiencia por el usuario	1	3	3	<i>Algunos roles necesitan estar relacionados con el sistema para su mejor funcionamiento.</i>

T4	Proceso interno complejo	1	1	3	<i>El sistema no posee cálculos complejos, pues es un sitio que se encarga de la divulgación de la información y de la interacción de sus usuarios.</i>
T5	Reusabilidad	1	0	2	<i>No es objetivo hacer reusabilidad del código,</i>
T6	Facilidad de instalación	0.5	1	0.5	<i>Por la naturaleza del sistema desarrollado no resulta compleja su instalación, ya que el mismo es Web.</i>
T7	Facilidad de uso	0.5	5	2.5	<i>El sistema debe ser fácil de usar ya que estará disponible para todo tipo de usuario.</i>
T8	Portabilidad	2	5	10	<i>El sistema debe ser portable ya que si se necesita cambiar de sistema operativo o de servidor este se debe mantener en funcionamiento.</i>
T9	Facilidad de cambio	1	5	5	<i>El sistema encuentra estructurada para que los cambios realizados afecten lo menos posible las funcionalidades del sistema.</i>
T10	Concurrencia	1	5	5	<i>La concurrencia es tratada con suma importancia.</i>
T11	Objetivos especiales de seguridad	1	5	5	<i>La seguridad del sistema es un tema bastante controlado, ya que el sistema sólo permite que un usuario realice las</i>

					<i>funcionalidades correspondientes a su rol dentro del sitio.</i>
T12	Acceso directo a terceras partes	1	2	2	<i>La aplicación es accesible a cualquier usuario.</i>
T13	Facilidades especiales de entrenamiento a usuarios finales	1	1	1	<i>No se hace necesario el entrenamiento de los usuarios finales, debido a la facilidad de uso que presenta el sistema, pero se debe incluir un manual de usuario para garantizar la correcta usabilidad de dicho sistema.</i>
			<i>Total Factor</i>	39	

El Factor de complejidad técnica se calcula mediante la siguiente ecuación:

$$TCF = 0.6 + 0.01 * \sum(\text{Peso}_i \times \text{Valor asignado}_i)$$

$$TCF = 0.6 + 0.01 * 39$$

$$TCF = 0.99$$

3.4.3.2 Determinación del factor ambiente

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente.

Tabla 3.4 Factores de ambiente.

Número del factor	Descripción	Peso	Valor	Factor	Comentario

E1	Familiaridad con el modelo del proyecto usado.	1.5	3	4.5	<i>Se está familiarizado con el modelo del proyecto, pero la experiencia en el modelado es media.</i>
E2	Experiencia en la aplicación	0.5	3	2	<i>No es una aplicación que requiera de mucha experiencia, pero se necesita de un equipo capacitado y de conocimientos suficientes para garantizar su correcto funcionamiento.</i>
E3	Experiencia OO.	1	4	4	<i>Se considera cierto grado de experiencia en la programación orientada a objetos (OO), debido a que esta es la que se ha impartido durante la carrera.</i>
E4	Capacidad del analista líder.	0.5	3	1.5	<i>No existe analista líder, ya que no existe equipo de desarrollo.</i>
E5	Motivación.	1	5	5	<i>Exista gran motivación para el desarrollo del proyecto.</i>
E6	Estabilidad de los requerimientos.	2	4	8	<i>Aunque el sistema se encuentra sujeto a cambios, el mismo brinda las funcionalidades esenciales que dan cumplimiento a los</i>

					<i>objetivos que iniciaron su realización.</i>
E7	Personal media jornada.	-1	0	0	<i>Se trabajará a tiempo completo.</i>
E8	Dificultad en lenguaje de programación.	-1	3	-3	<i>Como el lenguaje empleado fue PHP y este ofrece grandes facilidades y ventajas, se considera una dificultad media su empleo.</i>
			<i>Total</i>	<i>21.8</i>	

El factor de ambiente se calcula mediante la siguiente ecuación:

$$EF = 1.4 - 0.03 * \sum (\text{Peso}_i \times \text{Valor asignado}_i)$$

$$EF = 1.4 - 0.03 * 21.8$$

$$EF = 0.75$$

Cálculo de los Puntos de Casos de Uso Ajustados:

$$UCP = UUCP * TCF * EF$$

$$UCP = 117 * 0.99 * 0.75$$

$$UCP = 86.87$$

3.4.4 Calculo del esfuerzo

El esfuerzo en horas-hombre viene dado por:

$$E = UCP * CF \text{ donde:}$$

E: esfuerzo estimado en horas-hombre.

UCP: Puntos de casos de uso ajustados.

CF: Factor de conversión (20 horas-hombre por defecto).

$$E = 86.87 * 20$$

$$E = 1737.4$$

Para la obtención de una estimación más exacta de la duración del proyecto, se hace necesario agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las restantes actividades que se llevaron a cabo durante el desarrollo del software; así como la distribución del esfuerzo entre dichas actividades está dada por la siguiente aproximación:

Tabla 3.5 Distribución genérica del esfuerzo.

Actividad	Porcentaje
Análisis	10.00%
Diseño	20.00%
Programación	40.00%
Pruebas	15.00%
Sobrecarga(otras actividades)	15.00%

Con este criterio y tomando como entrada la estimación de tiempo calculada a partir de los Puntos de Casos de Uso, se pueden calcular las demás estimaciones para obtener la duración total del proyecto.

Tabla 3.6 Distribución real del esfuerzo.

Actividad	Porcentaje
Análisis	434.4

Diseño	868.7
Programación	1737.4
Pruebas	651.5
Sobrecarga(otras actividades)	651.5
<i>Total</i>	<i>4343.5</i>

Cálculo del esfuerzo total:

$$ETotal = 4343.5 \text{ horas /hombre}$$

Cálculo del tiempo de desarrollo:

$$TDesarrollo = ETotal/CHTotal \quad CHTotal: \text{ Cantidad de hombres}$$

$$TDesarrollo = 4343.5/1$$

$$TDesarrollo = 4343.5 \text{ horas}$$

Considerando que se trabajan 8 horas diarias:

$$TDesarrollo = TDesarrollo/8 \text{ horas/día}$$

$$TDesarrollo = 4343.5 \text{ horas}/8 \text{ horas/día}$$

$$TDesarrollo = 543 \text{ días aproximadamente}$$

Cálculo del costo:

$$CostoTotal = ETotal * 1 * TH \quad TH: \text{ Tarifa horaria}(= 1.031)$$

$$CostoTotal = 4343.5 * 1 * 1.031$$

$$CostoTotal = 4478.15$$

3.5 Beneficios tangibles e intangibles

El desarrollo de esta aplicación representa para el centro una serie de beneficios, tanto tangibles como intangibles, los cuales enuncian a continuación:

Beneficios tangibles:

Antes de poner en marcha el desarrollo de este software, se realizó un profundo estudio de las necesidades que conllevaban a su realización; entre las mismas se encontraba el ahorro de innumerables materiales, entre los que se encuentran: hojas, pegamento, marcadores permanentes y no permanentes, pintura de acuarelas entre muchos otros. Además la carencia de estos materiales en ocasiones representa una barrera para el desarrollo de las tareas, por ejemplo si después de confeccionada la cartelera y puesta en la puerta del centro cultural aparece una actividad no planificada entonces habría que borrar la anterior o pegar sobre ella un parche de papel actualizando la cartelera con la nueva actividad y a su vez las personas que ya habían consultado la cartelera una vez no se enterarían del cambio realizado, un sistema que garantice el buen funcionamiento, no sólo agilizaría el trabajo a realizar, sino que el ahorraría al Centro.

Beneficios intangibles:

Los beneficios intangibles son aquellos que no son del tipo material, son los beneficios inmateriales los que se esperan que sean más logrados con el desarrollo de esta aplicación, ya que su puesta en marcha traerá consigo una mejor una mejor comunicación entre el “Mejunje de Silverio”, el centro provincial de cultura, y todas las personas interesadas en el complejo cultural de forma más interactiva y eficiente.

Estos beneficios se detallan a continuación:

- Los usuarios emplearán con mayor frecuencia la navegación Web consultar la cartelera y artículos de interés publicados, manteniéndose así actualizados con las nuevas tecnologías.
- Aumentará la confiabilidad de la información, ya que esta se mantendrá almacenada en una base de datos, y estará publicada las 24 horas del día para su consulta.

3.6 Análisis de costos y beneficios

Al valorar desarrollo del software propuesto se tuvo en cuenta la no existencia de una única herramienta que fuera capaz de brindar las funcionalidades esenciales que

conllevaron al desarrollo de este sistema, o sea, que permitiera publicar de manera permanente la cartelera y artículos asociados a alguna actividad realizada; por lo que teniendo en cuenta los beneficios que se obtendrán con el desarrollo de la herramienta propuesta se determinó que su puesta en marcha representa ventajas y ahorros económicos para el centro cultural.

3.7 Conclusiones.

Considerando los resultados arrojados respecto a la factibilidad del software después del estudio realizado en este epígrafe, se concluye que brinda suficientes beneficios para cubrir sus costos, o sea, que se apoya la realización del sistema en el centro cultural, ya que es factible y económico; el mismo implicará un esfuerzo total de 4343.5 horas/hombre, para un tiempo de desarrollo de 543 días aproximadamente, se contarán con 1 hombre para su realización, lo que implica un costo de \$4478.15 para una tarifa horaria de \$1.031.

Conclusiones.

- El estudio de las necesidades del cliente y los procesos que se realizan manualmente son imprescindibles para el desarrollo de un sistema informático.
- El empleo de una metodología para el desarrollo de una aplicación informática es vital para lograr el objetivo final, el producto, y en él rasgos como la calidad.
- El sistema resultante está provisto de un ambiente cómodo, fácil de manejar y cumple con los estándares de diseño para una aplicación Web.
- En la fase de implementación se le dio a cada uno de los objetivos trazados al inicio del desarrollo de este trabajo, o sea se implementaron todos los servicios para brindar un control riguroso y organizado de toda la información referente al centro cultural “El Mejunje de Silverio” y a los usuarios relacionados con los mismos.
- Se desarrolló el sitio web del mejunje, el cual brinda la posibilidad de gestionar la información de la cartelera y la publicación de artículos vinculados a una actividad realizada en el centro, lo que posibilita una mayor comunicación entre “El Mejunje de Silverio”, el centro provincial de cultura, y todas las personas interesadas en el complejo cultural. Además se implementó un servicio web el cual podrá usarse para compartir en otras webs la cartelera
- Se estudió la factibilidad del desarrollo de este sistema, obteniendo el tiempo de desarrollo, el costo y los beneficios que va a reportar , lo que permitió determinar que el sistema es factible, pues ofrece una serie de beneficios, tanto tangibles como intangibles para la entidad en que se desarrolla. Además se hicieron pruebas de calidad para comprobar la eficacia del sistema realizado.

Recomendaciones.

Una vez desarrollado el sistema y cumplidos los objetivos propuestos en el trabajo se recomienda:

- Que el sistema sea probado, para comprobar si cumple con las expectativas y necesidades de los usuarios a la hora de manejar la información referente a la cartelera.
- Que los servicios web sean probados e integrados en varios sitios web de la provincia.
- Implementar versiones posteriores en caso que se le adicione alguna funcionalidad nueva al sistema.

Referencias bibliograficas

- AUTORES, G. D. 2009. Servicios Web: SOAP. *Unam*.
- CATALANI, E. 2008. *ARQUITECTURA Modelo/Vista/Controlador* [Online]. Available: www.exequielc.wordpress.com [Accessed 5-6-2012].
- CEIS., E. D. 2007. Requisitos del Sistema.
- ECURED. 2009a. *Calidad de software, tipos de pruebas* [Online]. [Accessed 22-6-2012].
- ECURED. 2009b. *PHP* [Online]. [Accessed 6-4-2012].
- ECURED. 2011. *CASE* [Online]. [Accessed 20-4-2012].
- FOUNTES, C. 2008. *Protocolos de Acceso* [Online]. Available: www.monografias.com [Accessed 1-6-2012].
- GOMEZ, A. 2010. *SOAP* [Online]. Available: www.w3c.es [Accessed 1-6-2012].
- GSP. 2002. *Manual Reference Pages - DUMMYNET* [Online]. Available: www.info.iet.unipi.it [Accessed 15-6-2012].
- JACOBSON, I. *El Proceso Unificado de Desarrollo de Software*.
- KEEKER, K. 2009. *Calidad del contenido en el Diseño de sitios Web atractivos*.
- PERALTA, M. Estimación del Esfuerzo Basada en Casos de Uso
- RAMIREZ, J. 2010. Programacion rapida con CakePHP. *Linux*.
- VISCAINO, A. 2008. Una herramienta CASE: Visual Paradigm.
- WIKIPEDIA. 2011a. *Modelo Vista Controlador* [Online]. [Accessed 17-5-2012].
- WIKIPEDIA. 2011b. *PHP* [Online]. [Accessed 6-4-2012].
- WIKIPEDIA. 2011c. *Servicios Web* [Online]. [Accessed 1-6-2012].
- WIKIPEDIA. 2011d. *SQL* [Online]. [Accessed 18-5-2012].
- ZAPATA, G. 2009. Herramientas de ISW.

