

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**FACULTAD # 8**

**ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN WEB  
“JULIO ANTONIO MELLA”**



**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE  
INGENIERO DE CIENCIAS INFORMÁTICAS**

Autores: María de los Ángeles Santos Salgado.  
Ariam Ajete Mena.

Tutor: Ing. Yordan Ernesto Estrada Rodríguez.  
Co-Tutor: Lic. En C/S Roberto López Dosagues.

Ciudad de la Habana, Junio 2009  
Año del 50 Aniversario de la Revolución

# DECLARACIÓN DE AUTORÍA

---

## DECLARACIÓN DE AUTORÍA

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

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_ del año

\_\_\_\_\_.

Autores:

María de los Ángeles Santos Salgado

Ariam Ajete Mena

\_\_\_\_\_

\_\_\_\_\_

Tutor:

Ing. Yordan Ernesto Estrada Rodríguez.

\_\_\_\_\_

Co-Tutor:

Lic En C/S Roberto López Dosagues.

\_\_\_\_\_





*“En el mundo ha de haber cierta cantidad de luz, como ha de haber cierta cantidad de decoro. Cuando hay muchos hombres sin decoro, hay siempre otros que tienen en sí el decoro de muchos hombres.”*

*José Martí*

### DEDICATORIA

*Mery:*

*En primer lugar le dedico mi trabajo a mi madre por demostrarme a cada instante el valor verdadero del sacrificio y ser ejemplo de abnegación y guía en cada uno de mis pasos. Mami, te amo.*

*A mi abuelito Tomas que aún estando ausente sigue siendo uno de mis motores impulsores y estaría muy orgulloso de mí.*

*A mi abuelita que me ha mimado desde pequeña y ha sufrido cada minuto conmigo y ha disfrutado de mis alegrías.*

*A mi papá por estar siempre presente en los momentos que lo he necesitado a mi lado.*

*A mi hermana Cindy por continuar mis pasos y estar repleta de sencillez, a quien he admirado por sus valores y por su confianza.*

*A mi novio Frank por haber llegado a mi vida en un momento crucial y por darme la mano y apoyarme incondicionalmente.*

*A Yanelis por ser más que mi amiga y aunque no sigamos juntas será alguien especial. Recuerda que aquí estoy yo para lo que necesites y que siempre seré tu hermanita habanera.*

*A mis amistades más queridas por compartir conmigo experiencias únicas que nunca olvidare (Lizy, Héctor, Nory, Marlon, Anny, Michel, Anisley, Nidia, Ernesto, Lisdey).*

*Y a todas las personas que influyeron mucho en mi vida durante este tiempo de estudio aquí en la Universidad.*

*Ariam:*

*Dedico este trabajo a mi madre que es como un todo en mi vida, es mi razón de ser y lo más grande que tengo en la vida.*

*A mi padre, al cual le debo el 95% de lo que soy y lo que seré, me ha servido de faro y guía en todo momento.*

*A mi hermana por estar siempre apoyándome y dándome fuerza en los momentos más duros de mi vida y específicamente en los de mi carrera como informático.*

*A mi novia Annia la cual me ha dado fuerza para terminar esta carrera y servirme de apoyo fuente de inspiración en el quehacer día a día.*

*En general dedico este trabajo a todas esas personas que confiaron en mí en todo momento, a mi familia, a mis amigos especialmente a Alexei, Roman, Boris, Yoisell, Rodolkis, Héctor, Yan, Alice, Lisandra y a mi gran conciencia Marlon.*

### *Mery*

*A mi tutor Yordan Ernesto Estrada por ayudarme y guiarme durante el proceso de desarrollo de mi trabajo de diploma y aguantar mis malcriadeces.*

*A Lizy por demostrarme que la verdadera amistad es como la fosforescencia, mientras más oscuro es el momento, mejor se ve. A ti gracias por ayudarme en uno de los instantes en que creí que la batalla estaba perdida y por creer en mí.*

*Al profesor Roberto López Dosagues por ser factor fundamental en mi estancia dentro de la UCI y por guiarme en diferentes ocasiones.*

*Al 8503 por todo lo que pasamos juntos, un beso grande.*

### *Ariam*

*Al profesor Desagües por confiar en nosotros aun cuando estaba en dudas la continuidad de nosotros en la UCI, proponiéndonos un tema de tesis y aceptando ser nuestro Co-Tutor.*

*A Yordan un recién graduado al cual acudimos para que fuera nuestro Tutor y nunca titubeó en dar un “sí”, el cual tampoco dio un “no” cada vez que acudimos a él en busca de ayuda o a la hora de resolver algún problema de trabajo.*

*A mi novia Annia la cual rió conmigo en los momentos de felicidad y lloró en los de angustia y desespero.*

*A mis compañeros de aula por estar siempre dispuestos a dar un paso al frente y brindar una mano amiga y salvadora en el momento adecuado.*

*También a mi compañera de tesis por ser la persona más desesperada, responsable e intransigente que he conocido en esta universidad, creo que sin ella este trabajo no hubiese salido en tiempo ni con calidad.*

### AGRADECIMIENTOS COMUNES

*A nuestro Comandante en Jefe Fidel Castro por haber creado la primera Universidad de la batalla de ideas y brindarnos la oportunidad de estudiar en este centro de referencia nacional.*

*A la Universidad de las Ciencias Informáticas por formarnos como profesionales y cubanos revolucionarios.*

*A nuestros profesores que durante todos estos años de estudio nos apoyaron y ayudaron.*

*Al todas las personas que de una manera u otra nos ofrecieron asistencia en el proceso de desarrollo del trabajo de diploma (Profesor Yasmani Ceballos, Alfredo Poll, Profesor Dionicio Estrada).*

*Gracias*

### RESUMEN

La Universidad de las Ciencias Informáticas en semejanza con otros centros de la educación superior pretende hacer una alta difusión de la labor revolucionaria de aquellas personas de la historia cubana que han tenido un alto significado para el país. En función de tales propósitos la dirección de la Universidad creó una Cátedra Mella que contribuye con la divulgación de toda una labor digna de reconocimiento como lo fue la de Julio Antonio Mella, alto dirigente de la Federación Estudiantil Universitaria y con el cual los jóvenes de hoy en día se ven identificados. El presente trabajo que lleva por título Análisis, Diseño e Implementación de la Aplicación Web Julio Antonio Mella pretende brindar la propuesta de creación de un sistema que contribuya dentro de la Universidad y junto a la Cátedra Mella llevar a cabo toda la exposición de la vida revolucionaria de tan importante figura cubana. Se pretende que el sistema proporcione los materiales suficientes y necesarios para lograr el propósito deseado y permitir a todo usuario que interactúe con la aplicación, tener un espacio donde se pueda intercambiar con el trabajo realizado por el destacado joven revolucionario. En el documento se recoge todo el proceso de investigación y elaboración de la aplicación que estuvo reflejado en el estudio del estado del arte, en la definición de las características a presentar el sistema además del proceso de planificación, diseño, implementación y prueba; todos abarcados de manera tal que se pudiera lograr la evolución y terminación del producto.

## Índice de Contenido

Introducción .....	1
Capítulo 1. Fundamentación Teórica.....	5
Introducción .....	5
1. Aplicación Web.....	5
1.1 Consideraciones técnicas.....	6
2. Arquitectura de software.....	6
2.1 Arquitectura de tres capas.....	7
2.2 Arquitectura cliente servidor.....	8
3. Metodologías de desarrollo .....	9
3.1 Metodología XP .....	10
3.2 Metodología RUP .....	21
3.3 Diferencias entre RUP y XP.....	22
4. Lenguaje de Programación Web .....	24
4.1 Lenguaje PHP .....	24
4.2 Java Script.....	27
5. AJAX.....	27
6.Wamp .....	28
7. Sistemas de Gestión de Contenido .....	29
7.1 Ejemplos de CMS.....	30
8. Sistemas Gestores de base de datos .....	34
8.1 PostgreSQL .....	34
9. Propuesta para el desarrollo.....	35
10. Conclusiones.....	36
Capitulo 2. Características del sistema.....	37

2.1 Introducción .....	37
2.2 Objeto de automatización. ....	37
2.3 Propuesta del sistema. ....	37
2.3 Flujo del sistema. ....	39
2.4 Personas relacionadas con la aplicación. ....	39
2.5 Conclusiones.....	41
Capitulo 3. Exploración y planificación .....	42
3.1 Introducción .....	42
3.2 Fase exploración. ....	42
3.3 Fase de Planificación. ....	49
3.4 Conclusiones.....	53
Capitulo 4. Diseño, desarrollo y prueba .....	54
4.1 Introducción .....	54
4.2 Diseño .....	54
4.3 Desarrollo. ....	57
4.4 Pruebas .....	67
4.5 Conclusiones.....	80
Conclusiones Generales.....	81
Referencias Bibliográficas .....	83
Bibliografía .....	84
Glosario de Términos .....	86

### ***Índice de Figuras***

Figura 1 Arquitectura Tres Capas. ....	8
Figura 2 Esquema del funcionamiento de la Arquitectura Cliente-Servidor .....	9
Figura 3 Ciclo de vida de RUP .....	22
Figura 4 Funcionamiento del lenguaje PHP. ....	26

### ***Índice de Tablas***

Tabla 1 Personal asociado a la aplicación. ....	40
Tabla 2 HU Crear Cuenta de Usuario. ....	42
Tabla 3 HU Mostrar Perfil de Usuario.....	43
Tabla 4 HU Modificar Perfil de Usuario. ....	43
Tabla 5 HU Autenticar Usuario.....	44
Tabla 6 HU Eliminar Cuenta de Usuario.....	44
Tabla 7 HU Publicar Materiales.....	45
Tabla 8 HU Eliminar Materiales.....	45
Tabla 9 HU Descargar Publicaciones.....	46
Tabla 10 HU Gestionar Noticias. ....	46
Tabla 11 HU Buscar Información. ....	46
Tabla 12 HU Gestionar Foro. ....	47
Tabla 13 HU Gestionar Encuesta.....	47
Tabla 14 Mostrar información.....	48
Tabla 15 Estimación de esfuerzo por HU. ....	49
Tabla 16 Plan de duración de iteraciones para el equipo de trabajo. ....	51
Tabla 17 Módulos e Historias de Usuarios Abarcadas. ....	52
Tabla 18 Plan de duración de entrega. ....	53
Tabla 19 Tarjeta CRC del Módulo Usuario.....	55
Tabla 20 Tarjeta CRC del Módulo Administrador ....	55
Tabla 21 Tarjeta CRC del Módulo Material. ....	56
Tabla 22 Tarjeta CRC del Módulo Perfil.....	56
Tabla 23 Tarjeta CRC del Módulo Noticia ....	57
Tabla 24 Tarjeta CRC al Módulo Listado.....	57
Tabla 25 Módulos Abordados en la Primera Iteración. ....	58
Tabla 26 Tarea 1 al Módulo Usuario ....	59
Tabla 27 Tarea 2 al Módulo Usuario. ....	59
Tabla 28 Tarea 1 al Módulo Material.....	60
Tabla 29 Tarea 2 al Módulo Material.....	60
Tabla 30 Módulos Abordados en la Segunda Iteración ....	61
Tabla 31 Tarea 1 al Modulo Administrador.....	61

Tabla 32 Tarea 2 al Modulo Administrador.....	62
Tabla 33 Tarea 3 al Modulo Material.....	62
Tabla 34 Tarea 3 al Módulo Usuario.....	63
Tabla 35 Tarea 1 al Modulo Perfil.....	63
Tabla 36 Tarea 1 al Módulo Noticia.....	64
Tabla 37 Tarea 2 al Módulo Noticia.....	64
Tabla 38 Tarea 3 al Módulo Noticia.....	65
Tabla 39 Tarea 1 al Módulo Listado.....	65
Tabla 40 Tarea 2 al Módulo Listado.....	66
Tabla 41 Módulos Abordados en la Tercera Iteración.....	66
Tabla 42 Tarea 3 al Módulo Usuario.....	66
Tabla 43 Tarea 3 al Módulo Administrador.....	67
Tabla 44 Prueba 1 al Módulo Administrador.....	69
Tabla 45 Prueba 2 al Módulo Administrador.....	69
Tabla 46 Prueba 3 al Módulo Administrador.....	70
Tabla 47 Prueba 4 al Módulo Administrador.....	71
Tabla 48 Prueba 5 al Módulo Administrador.....	72
Tabla 49 Prueba 1 al Módulo Usuario.....	72
Tabla 50 Prueba 2 al Módulo Usuario.....	73
Tabla 51 Prueba 3 al Modulo Usuario.....	74
Tabla 52 Prueba 4 al Módulo Usuario.....	74
Tabla 53 Prueba 1 al Módulo Material.....	75
Tabla 54 Prueba 2 al Módulo Material.....	76
Tabla 55 Prueba 1 al Módulo Perfil.....	77
Tabla 56 Prueba 2 al Módulo Perfil.....	77
Tabla 57 Prueba 1 al Modulo Noticias.....	78
Tabla 58 Prueba 2 al Módulo Noticias.....	79
Tabla 59 Prueba 1 al Módulo Listado.....	79

## Introducción

En la actualidad las aplicaciones Web han cambiado la cultura de una manera considerable, han sido una nueva Revolución tecnológica que ha tomado fuerzas y que lleva consigo una formidable expansión a nivel mundial. Se han popularizado por los elementos que las caracterizan y que permiten una interesante comunicación entre el usuario y la información.

En Cuba, la educación constituye uno de los máximos logros alcanzados después del triunfo revolucionario, poder informatizarla logrando un nivel superior de desarrollo y calidad en tan importante esfera de la sociedad, se ha convertido en un objetivo de alta prioridad. Alcanzar la asociación de la historia cubana con aplicaciones Web que viabilicen divulgar temas históricos que profundicen conocimientos y amenicen el estudio de la historia, es una combinación perfecta que se transforma en una potente herramienta dentro de la enseñanza y colabora con el acercamiento a otras épocas y figuras importantes para Cuba.

Una de las personalidades cimeras de la historia cubana lo fue Julio Antonio Mella quien constituye un líder indiscutible del estudiantado universitario y comunista comprometido con la defensa de los ideales más puros del pueblo cubano. En él se resume una de las personalidades revolucionarias más íntegras y poderosas de su tiempo. Su misión de vanguardia, en las nuevas circunstancias históricas aparecidas con la República neocolonial, lo llevaron a la creación de instrumentos para la lucha que se avecinaba. Ejemplo real de líder comunista se caracterizó por su lucha constante e incansable quehacer revolucionario encontrando como su primer campo de batalla las aulas de la Universidad de La Habana siendo un terreno donde se destacó como organizador y conductor de masas.

Los universitarios cubanos encuentran en hombres como lo fue Julio Antonio Mella, el más alto ejemplo de integralidad y abnegación, constituye parte de la élite estudiantil cubana y que hoy merece un cándido homenaje por parte de las posteriores generaciones que siguen su legado político. Es por ello que con la implementación de una aplicación Web que permita la socialización de la labor revolucionaria de Julio Antonio Mella se establece una vía de concretizar la idea de convertir el empleo de la informática y sus elementos en una poderosa vertiente de conocimientos aplicada a la educación con el principal objetivo de engrandecer la cultura de todo un pueblo altamente identificado con su historia.

### Situación Problémica:

La Universidad de las Ciencias Informáticas cuenta con una Cátedra Mella que se encarga de promover a través de un trabajo educativo profesional, la vida de un ilustre revolucionario cubano como lo fue Julio Antonio Mella. A pesar de los numerosos materiales que contribuyen con esta labor entre los que se destacan: libros, artículos , videos, fotografías, documentación tangible rica en conocimiento, la Cátedra no se apoya en un producto informático que le facilite la socialización, divulgación y búsqueda de información sobre el pensamiento y obra del joven revolucionario, entre los miembros de la comunidad UCI y el resto de los universitarios de Cuba y del mundo, que constituya una guía para la acción transformadora de la sociedad.

### Problema a resolver:

Necesidad de un producto informático que divulgue el pensamiento revolucionario de Julio Antonio Mella y sea fuente de conocimientos para todo personal que tenga acceso a la información que se brinde.

### Objetivo General:

Desarrollar una aplicación Web encaminada a divulgar el pensamiento revolucionario de Julio Antonio Mella y la labor política de tan destacada figura de nuestra historia.

### Objeto de Estudio:

Proceso de desarrollo de una aplicación Web que brinde la posibilidad de difundir el pensamiento de Mella.

### Campo de Acción:

Vida y obra de Julio Antonio Mella.

### Ideas a Defender:

Si se desarrolla una aplicación Web sobre Julio Antonio Mella entonces se lograría mayor divulgación de la labor política y revolucionaria del líder estudiantil posibilitando trascender los

marcos de la comunidad UCI y permitir a todos los que accedan a la aplicación, conocer el legado político de Mella.

### Objetivos Específicos:

- ✓ Investigar acerca del proceso revolucionario de Julio Antonio Mella y hacer un seguimiento de su trabajo político para lograr una mayor profundización y eficacia en la aplicación que se pretende implementar.
- ✓ Efectuar el estudio de las características de las herramientas que se utilizaran para desarrollar e implementar la aplicación.
- ✓ Plantear e implementar el diseño de interfaz gráfica de la aplicación Web.
- ✓ Elaborar el texto teórico de la investigación.
- ✓ Realizar el proceso de desarrollo de la aplicación Web Julio Antonio Mella.
- ✓ Validar el producto obtenido al concluir su implementación.

### Tareas de Investigación:

- ✓ Realizar entrevistas a los miembros de la cátedra Mella para obtener información a utilizar en la posterior implementación de la aplicación.
- ✓ Búsqueda de los medios para la ejecución de la aplicación.
- ✓ Consultar bibliografías sobre las herramientas informáticas a utilizar para el desarrollo de la aplicación.
- ✓ Efectuar un levantamiento de los requisitos que debe cumplir la aplicación para llegar a obtener el resultado final esperado donde se cumplan las expectativas de los usuarios que la visiten.
- ✓ Realizar la planificación de las tareas a realizar para la futura implementación del sistema.
- ✓ Realizar pruebas al sistema para comprobar las funcionalidades que lo caractericen.

El trabajo está estructurado en cuatro capítulos de los cuales se realiza una descripción a continuación:

En el primer capítulo se fundamenta sobre las aplicaciones Web y se hace referencia a sus tendencias. Se explican y exponen las herramientas posibles a emplear en la concretización del sistema y se guía todo el proceso de desarrollo del capítulo en función de obtener una propuesta para la evolución del sistema a implementar.

En el capítulo 2 se plantean las características a presentar el sistema siendo reflejadas con la descripción del flujo de trabajo, el objeto de automatización y las personas relacionadas a éste.

El desarrollo del tercer capítulo se guía entorno a la fase de Exploración y Planificación haciendo alusión a las Historias de Usuario que reflejan las características funcionales que ha de tener la aplicación y la estimación del esfuerzo a emplear en cada una. Se manifiesta el plan de iteraciones así como los módulos declarados dentro del sistema para ordenar el trabajo de implementación de la aplicación.

El capítulo 4 detalla los aspectos relacionados al diseño, la implementación y las pruebas del sistema. En este se mencionan las tarjetas CRC junto a las tareas de implementación que se efectuaron en esta fase y las pruebas realizadas por parte del cliente comprobando el buen funcionamiento del sistema. Con la culminación del capítulo planteado se espera haber alcanzado las metas trazadas por parte del equipo de desarrolladores de la aplicación.

### **Introducción**

Las aplicaciones Web han tenido una alta y significativa demanda como producto de las transformaciones tecnológicas que han ocurrido desde hace unos años y por el amplio uso que estas pueden tener en diferentes ramas de la sociedad posibilitando cambios económicos, políticos, tecnológicos y sociales.

En la Universidad de las Ciencias Informáticas se han implementado aplicaciones Web que sirven no solo para manejar datos de información sino para crear un ambiente de divulgación de diferentes temas que sean del interés de la comunidad universitaria y del resto de las personas que quieran acceder a estas aplicaciones. En función de ofrecerle a la Cátedra Mella un medio de divulgación informático sobre la labor de Julio Antonio Mella, se realizará en este capítulo un estudio de aquellas tecnologías y herramientas posibles de explotar en una futura implementación.

### **1. Aplicación Web**

En la ingeniería del software se denominada aplicación Web a todas aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web (HTML, Java Script, Java, etc.) en la que se confía la ejecución al navegador.

Las aplicaciones Web son muy populares debido a lo práctico del navegador Web como cliente ligero, así como a la facilidad para ser actualizadas y mantenidas sin distribuir e instalar software a miles de usuarios potenciales.

Una de las características relevantes a mencionar de las aplicaciones de este tipo es que permiten una relación y comunicación activa entre el usuario y la información brindando la oportunidad de que el usuario acceda a los datos de modo interactivo pues la aplicación

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

responderá a cada una de las acciones entre las que podemos enunciar: rellenar y enviar formularios o acceder a gestores de base de datos de todos tipos.

Las interfaces web tienen ciertas limitaciones en las funcionalidades que se ofrecen al usuario, por tales motivos es que en la actualidad, los desarrolladores web utilizan generalmente lenguajes interpretados o script en el lado del cliente para añadir más funcionalidades, especialmente para ofrecer una experiencia interactiva que no requiera recargar la página cada vez (lo que suele resultar molesto a los usuarios). Recientemente se han desarrollado tecnologías para coordinar estos lenguajes con tecnologías en el lado del servidor.

## 1.1 Consideraciones técnicas

Las aplicaciones Web deberían funcionar igual independientemente de la versión del sistema operativo instalado en el cliente, esto permite que en vez de tener la necesidad de crear clientes para Windows, Mac OS X, GNU/Linux, y otros sistemas operativos, la aplicación Web se escribe una vez y se ejecuta igual en todas partes. Sin embargo, hay aplicaciones inconsistentes escritas con HTML, CSS, DOM y otras especificaciones para navegadores web que pueden causar problemas en el desarrollo y soporte de las mismas.

Una consideración técnica destacable es utilizar Adobe Flash Player o Java applets para desarrollar parte o toda la interfaz de usuario. Como casi todos los navegadores incluyen soporte para estas tecnologías las aplicaciones basadas en Flash o Java pueden ser implementadas con aproximadamente la misma facilidad. Dado que ignoran las configuraciones de los navegadores, estas tecnologías permiten más control sobre la interfaz, aunque las incompatibilidades entre implementaciones Flash o Java puedan crear nuevas complicaciones. Por las similitudes con una arquitectura cliente-servidor, con un cliente "no ligero", existen discrepancias sobre el hecho de llamar a estos sistemas "aplicaciones Web"; un término alternativo es "Aplicación Enriquecida de Internet".

## 2. Arquitectura de software

Un planteamiento sobre la definición de la arquitectura de software fue expresado por IEEE Std 1471-2000 como:

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

“La Arquitectura del Software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución”. (1).

Tal fundamentación permiten reafirmar y describir la arquitectura de un software como el diseño de más alto nivel en la estructura de un sistema, también denominada estructura lógica que se basa en el conjunto de patrones y abstracciones que brinda el marco de referencia necesario para guiar y lograr la construcción del software.

La Arquitectura establece los fundamentos para que analistas, diseñadores, programadores, etc. trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades, es decir permite que el equipo de desarrolladores tenga una guía para llegar a obtener el producto final esperado.

## 2.1 Arquitectura de tres capas

La arquitectura de tres capas es un diseño reciente que introduce una capa intermedia en el proceso. Cada capa es un proceso separado y bien definido corriendo en plataformas separadas. En la arquitectura tradicional de tres capas se instala una interfaz de usuario en la computadora del usuario final (el cliente). La arquitectura basada en Web transforma la interfaz de búsqueda existente (el explorador de Web), en la interfaz del usuario final. (1)(2)

Capa de presentación: Esta capa es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario. Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" para el usuario generalmente se presentan como formularios.

Capa de negocio: En esta capa se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Capa de datos: Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

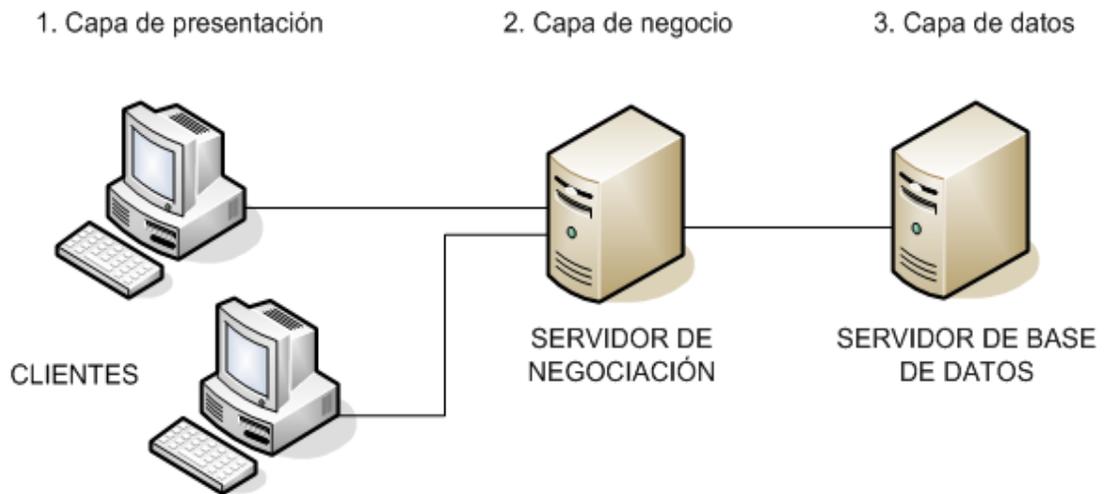


Figura 1 Arquitectura Tres Capas.

### 2.2 Arquitectura cliente servidor.

La arquitectura cliente -servidor se basa básicamente en un programa (cliente) que realiza ciertas peticiones a otro que le brinda respuestas (servidor)

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

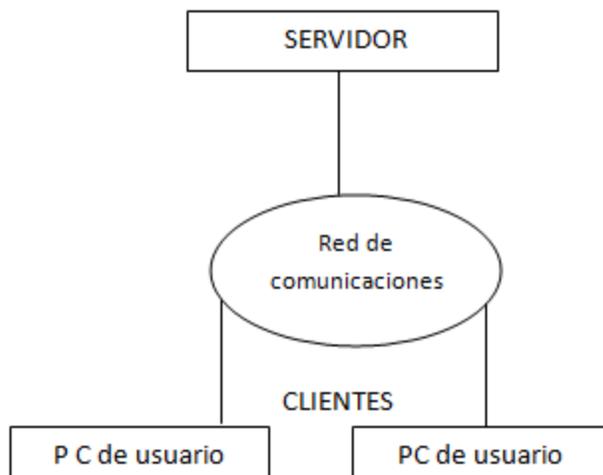


Figura 2 Esquema del funcionamiento de la Arquitectura Cliente-Servidor

### 3. Metodologías de desarrollo

Una metodología es aquella guía que se sigue para realizar las acciones propias de una investigación. Esta guía nos va indicando qué hacer y cómo actuar cuando se quiere obtener algún tipo de investigación. Es posible definir una metodología como aquel enfoque que permite observar un problema de una forma total, sistemática, disciplinada y con cierto orden.

En un artículo escrito por la Licenciada en Informática María A. Mendoza Sánchez se plantea: “Todo desarrollo de un software es riesgoso y difícil de controlar, pero si no llevamos una metodología de por medio, lo que obtenemos es clientes insatisfechos con el resultado y desarrolladores aun más insatisfechos”. (2)

No es menos cierto que la ausencia de una metodología es un riesgo dentro de un proceso de evolución para un proyecto, pues el concepto esencial de metodología implica las técnicas de investigación, las cuales se definen como aquellos procedimientos que se llevan a cabo para efectuar con eficiencia la aplicación de la metodología que sea escogida para guiar el desarrollo de una investigación.

Para darle solución al problema planteado, se realizó el estudio de dos de las metodologías más conocidas: RUP y XP. Ambas metodologías son consideradas robustas para guiar todo el proceso de creación que se espera poner en práctica. Para efectuar el estudio tanto de RUP

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

como de XP se hace necesario apoyarse en determinada cantidad de documentos bibliográficos y de personal que tenga plenos conocimientos acerca del tema y que puedan socializar todo tipo de información considerada relevante para la consumación del trabajo. (3)

### 3.1 Metodología XP

La metodología eXtreme Programming nace como una nueva disciplina de desarrollo de software hace aproximadamente seis años creada por un programador llamado Kent Beck quien afirmó: " Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando este tiene lugar." (3)

La idea de crear una metodología aplicable a proyectos donde puedan cambiar las circunstancias, constituyó una condición para que surgiera la metodología XP caracterizada por ser aplicable a sistemas con constantes transformaciones.

Esta metodología a partir de su aparecimiento ha causado un gran revuelo entre el colectivo de programadores a nivel mundial. XP constituye una metodología en la cual se acepta los cambios de requerimientos aún tardíos en el ciclo de desarrollo. Brinda la oportunidad de integrar gerentes, clientes y desarrolladores en la búsqueda de calidad en el software potenciando al máximo el trabajo en equipo. Introduce un cambio en la forma de programar y mantiene el diseño simple y claro. Con XP se logra la entrega temprana así como la implementación de los cambios al ir siendo sugeridos. No ahorra en hardware además de construir programas entendibles y extensibles. Permite automatizar las pruebas: escribe código de prueba antes, durante y después de la programación: antes, al definir la funcionalidad; durante, al descubrir errores; después, en la integración. Es considera una revalorización del software con mayor énfasis en la calidad. En fin la programación extrema se basa en la simplicidad, la comunicación, el reciclado continuo de código y el énfasis en las pruebas permitiendo mejoras al proyecto en cuanto a realimentación y emprendimiento.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

Los objetivos de XP son muy simples: el primero, la satisfacción del cliente. Esta metodología trata de darle al cliente el software que él necesita y cuando lo necesita. Es decir, se debe responder muy rápido a las necesidades del cliente, incluso cuando los cambios sean al final de ciclo de la programación.

El segundo objetivo fundamentando lo planteado anteriormente es potenciar al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el progreso del software.

Por tanto se emplea XP cuando los clientes no tienen idea clara de los requerimientos y los van cambiando. Para proyectos de riesgo: fecha fija de entrega, algo nunca hecho por el grupo, algo nunca hecho por la comunidad de desarrolladores. No es apto para proyectos con mucho personal. Integra gerentes y clientes a la formulación de preguntas, la negociación de cronograma y alcances, la creación de las pruebas. También se usa esta metodología cuando es necesario automatizar las pruebas; es posible en casi todos los dominios. Es autorizado repensar el diseño para facilitar el ensayo. El objetivo es entregar el software tal cual se necesita y en el momento en que se necesita. Incidentalmente, los proyectos XP muestran mayor productividad por las características que presenta la metodología.

La importancia de aplicar una metodología al proceso de desarrollo de una aplicación se hace evidente desde los comienzos del proyecto hasta la conclusión de este, siempre buscando alcanzar la calidad anhelada, pues no es en vano que se afirma que aunque una metodología no hace el trabajo por sí sola, podrá colaborar lo suficiente.

La metodología XP es una colección de reglas y prácticas mutuamente soportadas que usadas en conjunto la definen. (4)

### 3.1.1 Reglas y prácticas

El mayor beneficio de las prácticas se consigue con su aplicación conjunta y equilibrada ya que se apoyan unas en otras. La mayoría de las prácticas propuestas por XP no son novedosas sino que en alguna forma ya habían sido propuestas en ingeniería del software e incluso demostrado su valor en la experiencia para un análisis histórico de ideas que sirven como antecedentes a las utilizadas por las metodologías ágiles. El mérito de XP es integrarlas de una

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

forma efectiva y complementarlas con otras ideas desde la perspectiva del negocio, los valores humanos y el trabajo en equipo.

### Planificación.

- Se escriben los relatos de usuario ("user stories").
- El plan de entregas crea el cronograma.
- Entregas pequeñas muy frecuentes.
- Se mide la velocidad del proyecto.
- Se divide el proyecto en fases de iteración.
- La planificación de iteración inicia cada fase.
- El personal rota por las diferentes áreas del sistema.
- Reuniones de pie, todos los días.
- Corregir el XP si falla (cambiar las reglas por consenso).

### Diseño.

- Simplicidad.
- Elegir una "metáfora" de sistema, un conjunto de nombres ilustrativos de la realidad.
- Usar tarjetas CRC en las sesiones de diseño.
- Crear soluciones rápidas puntuales ("spike solutions") para reducir el riesgo.
- No agregar funcionalidad antes de tiempo.
- Refactorizar dónde y cuándo sea posible.
- Codificación.
- El cliente siempre al alcance.
- El código debe respetar las normas aceptadas.
- Codificar primero las pruebas de unidad (unit tests).
- Todo el código se escribe en parejas de programadores.
- Integra código sólo una pareja por vez.
- Integrar seguido.
- El código es propiedad colectiva.
- No optimizar hasta el final del proyecto.

- No trabajar horas extra.

### **Prueba.**

- Todo el código debe tener pruebas de unidad.
- Todo el código debe pasar las pruebas de unidad antes de la entrega.
- Crear pruebas toda vez que aparezcan errores.
- Correr frecuentes pruebas de aceptación y publicar la puntuación.

### **3.1.2 XP más detallado.**

#### **Relatos de usuario.**

- ✓ Similares a los Casos de Uso, pero no lo mismo; estiman el tiempo con vistas al plan de entrega.
- ✓ Las escriben los clientes, describen algo que el sistema debe hacer, son 2 o 3 oraciones en el lenguaje del usuario, sin jerga técnica, con poco detalle. La descripción detallada se dará al momento de iniciar el desarrollo.
- ✓ Definen una o más pruebas de aceptación.
- ✓ Corresponden a 1, 2 o 3 semanas de desarrollo a tiempo completo.
- ✓ En un plan de entregas típico, se definen unas 80 historias  $\pm$  20%.

#### **Planificación de entrega.**

- ✓ En una reunión de planificación se define un plan de entrega para todo el proyecto. El plan de entrega se usará para crear planes de iteración.
- ✓ Se toman decisiones de desarrollo y de negocios, se negocia un cronograma cumplible por todos.
- ✓ Se parte de los relatos de usuario, impresos o en tarjetas; moviéndolas en la mesa se define el primer (o el siguiente) conjunto a implementar, algo utilizable, verificable, con sentido para la empresa, entregable a corto plazo.
- ✓ Puede planearse por tiempo (cuántas historias antes de tal fecha) o por alcance (cuánto tiempo para estas historias), a la velocidad de programación estimada para el equipo.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

- ✓ Un proyecto puede ser medido en 4 variables: 1) Alcance: qué se hará; 2) Recursos: gente disponible; 3) Fecha de entrega; 4) Calidad: bien hecho, bien probado. Sólo pueden elegirse 3 de las 4; la 4a. resulta de las anteriores. Además, la calidad no debiera cambiarse.

### **El Plan de Entrega.**

- ✓ Define los conjuntos de relatos de usuario a implementar en cada iteración, y sus fechas de entrega.
- ✓ Cuando se verifica un cambio de la velocidad de programación durante 2 o 3 iteraciones deberá rehacerse el Plan de Entrega.

### **Entregas frecuentes.**

- ✓ El Plan de Entrega busca definir unidades funcionales con sentido para producir entregas frecuentes.

### **Velocidad de programación.**

- ✓ Es la cantidad de relatos de usuario o tareas de programación realizadas en una iteración (un simple conteo).
- ✓ Es inútil sacar medias por programador o por tarea; cada equipo tiene sus propias características.
- ✓ Se esperan variaciones en la velocidad. Si son extremas, rehacer el Plan de Entrega.

### **Desarrollo de una iteración.**

- ✓ Dividir el Plan de Entrega en unas 12 iteraciones, más o menos. Cada iteración demora de 1 a 3 semanas.
- ✓ No adelantar planificación: cada iteración se planifica inmediatamente antes de comenzar. No adelantar implementación, desarrollar sólo lo previsto en la presente iteración.

### **Plan de iteración.**

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

- ✓ Incluye los relatos de usuario y las pruebas de aceptación. Ambas se dividen en tareas de programación. Una tarea de programación abarca de 1 a 3 días de trabajo ideales, sin interrupciones ni otras actividades.
- ✓ El programador estima la tarea que toma.
- ✓ Estimar el tiempo de acuerdo a la velocidad de programación ya verificada. Agregar o quitar tareas si no alcanza el tiempo; si es preciso se renegociará. No engañarse ignorando o alterando el cálculo frío.
- ✓ Insistir en implementar primero lo esencial; si no se llega será más fácil renegociar lo accesorio.
- ✓ No sacrificar la refactorización ni las pruebas de unidad: descuidar esto atrasará más.

### **Rotación del personal.**

- ✓ Mover al personal de un área a otra para evitar pérdidas de conocimiento o cuellos de botella en la codificación; todos deben conocer bastante código de otras secciones. La rotación y la programación en parejas produce el entrenamiento cruzado que evita las "islas de conocimiento".
- ✓ En cada iteración todos deben trabajar parte del tiempo en algún área nueva. En cada pareja de programadores se rota sólo uno por vez.

### **Reunión diaria de pie.**

- ✓ Organizar una única reunión diaria, breve, de pie, con todos los integrantes, sólo para plantear problemas. Estos se resolverán luego en pequeños grupos de involucrados.

### **Corregir XP.**

- ✓ Adaptar las reglas cuando no se obtienen los resultados esperados. Las reglas se pueden cambiar pero no pueden dejar de existir: cada desarrollador debe saber exactamente qué puede esperar de los demás, y los demás de él.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

### **Buscar la simplicidad.**

- ✓ Un diseño simple es más fácil de crear y mantener. Buscar afanosamente el diseño más simple que funcione. No implementar funcionalidad antes de tiempo.
- ✓ Advertencia: lograr un diseño simple es un gran trabajo.

### **Esquema de nombres ("system metaphor").**

- ✓ Nombrar clases y métodos en forma consistente y clara. Debe ser posible inferir la significación real de un nombre nunca visto antes, a qué cosa se refiere.

### **Tarjetas CRC.**

- ✓ Usar tarjetas CRC, "Clase, Responsabilidad, Colaboración", en el diseño grupal. Ayudan a evitar el enfoque procedimental y destacan la orientación a objetos.
- ✓ Cada tarjeta CRC representa un objeto. El nombre de Clase va arriba, las responsabilidades (qué debe hacer) a la izquierda, las clases asistentes (que colaboran) a la derecha.
- ✓ No suele ser necesario escribir la tarjeta completa; los participantes se familiarizan rápidamente con el propósito de cada clase.
- ✓ En la reunión CRC alguien simula el sistema discutiendo los mensajes intercambiados entre objetos. Limitar a 1 o 2 personas de pie exponiendo, mientras los demás permanecen sentados.

### **Solución rápida ("spike solution").**

- ✓ Crear una solución rápida, un programa muy simple para explorar una solución posible, concentrada en el problema central sin atender el resto. Suele ser descartado; se emplea para probar la solución, mejorar la estimación, reducir el riesgo potencial.

### **Nunca agregar funcionalidad extra.**

- ✓ No agregar funcionalidades extra sólo porque ahora se ve claro cómo hacerlo, o porque parezca mejorar mucho el sistema: el 90% de las funcionalidades adicionales no llegan

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

a usarse. No prever requerimientos futuros ni flexibilidad extra. Concentrarse en el desarrollo de hoy.

### **Refactorizar a todo trance.**

- ✓ Refactorizar es mejorar el código existente, la estructura interna del software, no su comportamiento visible.
- ✓ Refactorizar para mantener el diseño simple: quitar redundancia, eliminar funcionalidad no usada, rehacer diseños obsoletos, mantener el código limpio y conciso para que sea fácil de entender, modificar y extender.

### **El cliente siempre presente.**

- ✓ El cliente debe participar activamente a lo largo de todo el proceso, el experto, no un ayudante. Debe haber una pareja de representantes del cliente asignadas al proyecto, a veces en tiempo completo.
- ✓ Los clientes escriben los relatos de usuario, participan en las estimaciones, asignan prioridades, verifican el cumplimiento de las funcionalidades. En la reunión de planificación de entrega negocian el conjunto de relatos de usuario a implementar en cada entrega.
- ✓ Los relatos de usuario no contienen detalles; los clientes deben estar presentes al definir las tareas de programación necesarias para implementar cada relato.
- ✓ Los clientes deben estar presentes en las pruebas de verificación para revisar el resultado y determinar cuándo puede pasarse a producción el sistema.

### **Normas de codificación.**

- ✓ Debe elegirse y respetarse una norma de codificación.

### **Codificar primero la prueba de unidad ("Unit Test").**

- ✓ Las pruebas de unidad se escriben una vez y se corren reiteradamente a lo largo de todo el proyecto, asegurando siempre el funcionamiento correcto; evitan las ambigüedades, los requerimientos quedan duros en la prueba.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

- ✓ Para cada unidad, se codifica primero una prueba simple para una función simple; se van agregando pruebas y funcionalidad en etapas sucesivas, hasta implementar todo y probar todo en esa unidad.
- ✓ Una funcionalidad está terminada cuando pasa todas sus pruebas de unidad.

### **Programación en parejas.**

- ✓ Todo el código incluido en una entrega de producción es escrito por dos personas trabajando juntas ante un único computador. Uno digita y piensa tácticamente sobre el método en construcción; el otro piensa estratégicamente en la integración de ese método en la clase.
- ✓ La programación en pareja produce la misma cantidad de código a un nivel de calidad superior.
- ✓ Lleva tiempo acostumbrarse a programar en pareja; puede resultar incómodo al principio.

### **Integración secuencial.**

- ✓ Los grupos trabajan en paralelo, pero integran uno por vez: sólo una pareja de programadores integra, prueba y entrega los cambios en el repositorio de código, en un momento dado. Allí se fija una nueva versión.
- ✓ Una forma de asegurar la integración secuencial es hacerla en una máquina única (si los programadores trabajan en un mismo lugar físico).
- ✓ Los programadores pueden integrar libremente en sus máquinas de trabajo habituales.

### **Integración frecuente.**

- ✓ Los programadores deben integrar su código al repositorio común cada pocas horas, al menos una vez por día cada pareja. Todos deben trabajar con la última versión.
- ✓ La integración continua evita los esfuerzos divergentes o fragmentados donde no se comunica lo que se puede compartir o reutilizar; las incompatibilidades se detectan temprano.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

### **Código de propiedad colectiva.**

- ✓ No hay una única persona responsable por el código: la construcción y la responsabilidad están distribuidas, todos los desarrolladores pueden agregar funcionalidad, corregir errores o refactorizar.
- ✓ El código incluye siempre las pruebas de unidad. El código sólo puede entregarse luego de haber pasado el 100 % de las pruebas de unidad.

### **Optimizar al final.**

- ✓ Dejar la optimización para el final. No suponer dónde estarán los puntos de estrangulamiento; medirlos.
- ✓ "Make it work, make it right, then make it fast." (Hágalo funcionar, hágalo bien, recién después hágalo andar rápido).

### **No trabajar horas extra.**

- ✓ El trabajo en horario extendido desmoraliza al equipo; no se cubrirá un atraso trabajando horas extra. Renegociar el alcance o la fecha de entrega en una reunión de planificación.
- ✓ Agregar personal a un proyecto atrasado es una mala idea.

### **Pruebas de unidad.**

- ✓ Crear y obtener un marco de prueba ("test framework") para crear una suite automática de pruebas de unidad. Probar todas las clases del sistema
- ✓ Crear las pruebas antes de escribir el código. No puede integrarse código sin sus pruebas de unidad.
- ✓ Las pruebas de unidad evolucionan junto con el código. No pueden crearse al final, ni dejar de escribirse.
- ✓ El tiempo de escribir las pruebas de unidad se gana con creces en la reiteración continua de las pruebas y la confianza al encarar cambios.
- ✓ Las pruebas de unidad posibilitan la propiedad colectiva de código, la refactorización, la integración frecuente. El agregado de funcionalidad incluye el agregado de pruebas.

### Marco de prueba.

- ✓ El marco de prueba no es una herramienta de prueba sino de desarrollo, se debe usar desde el principio. Puede crearse desde cero, pero los hay disponibles para la mayoría de los lenguajes

### Cuando aparece un error.

- ✓ Se crean pruebas para ese error, para evitar su reaparición.
- ✓ La aparición de un error en producción requiere crear una prueba de aceptación. Ante una prueba de aceptación fallida, los programadores crean pruebas de unidad para ubicar el defecto en el código. Una vez superadas el 100% de las pruebas de unidad se vuelve a correr la prueba de aceptación fallida para verificar la desaparición del error.

### Prueba de aceptación.

- ✓ Las pruebas de aceptación se crean a partir de los relatos de usuario. El cliente define los escenarios de prueba para verificar si el relato de usuario ha sido correctamente implementado. Un relato de usuario puede tener una o varias pruebas de aceptación. El cliente es responsable de verificar el pasaje de las pruebas de aceptación y priorizar la corrección de las pruebas fallidas.
- ✓ Las pruebas de aceptación son pruebas tipo caja negra a nivel del sistema: cada prueba de aceptación corresponde a un resultado producido por el sistema.
- ✓ Las pruebas de aceptación deben ser automáticas, correrse frecuentemente, publicarse sus resultados y programarse su corrección para la próxima iteración.
- ✓ Deben crearse pruebas de aceptación en cada iteración. Si no hay pruebas de aceptación nuevas no se ha hecho nada nuevo.
- ✓ Un relato de usuario no está completo hasta no haber pasado todas sus pruebas de aceptación.

### 3.2 Metodología RUP

RUP es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de usos, posee una forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo) al igual que pretende implementar las mejores prácticas en Ingeniería del Software. Con la aplicación de RUP se puede observar el desarrollo iterativo, la administración de requisitos, el uso de arquitectura basada en componentes y el control de los cambios. RUP incluye un modelado visual del software y una verificación de la calidad. (5)

#### 3.2.1 Ciclo de vida

El ciclo de vida RUP es una implementación del Desarrollo en espiral. El ciclo de vida organiza las tareas en fases e iteraciones.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades.

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una línea base de la arquitectura. Durante la fase de inicio las iteraciones hacen mayor énfasis en actividades de modelado del negocio y de requerimientos.

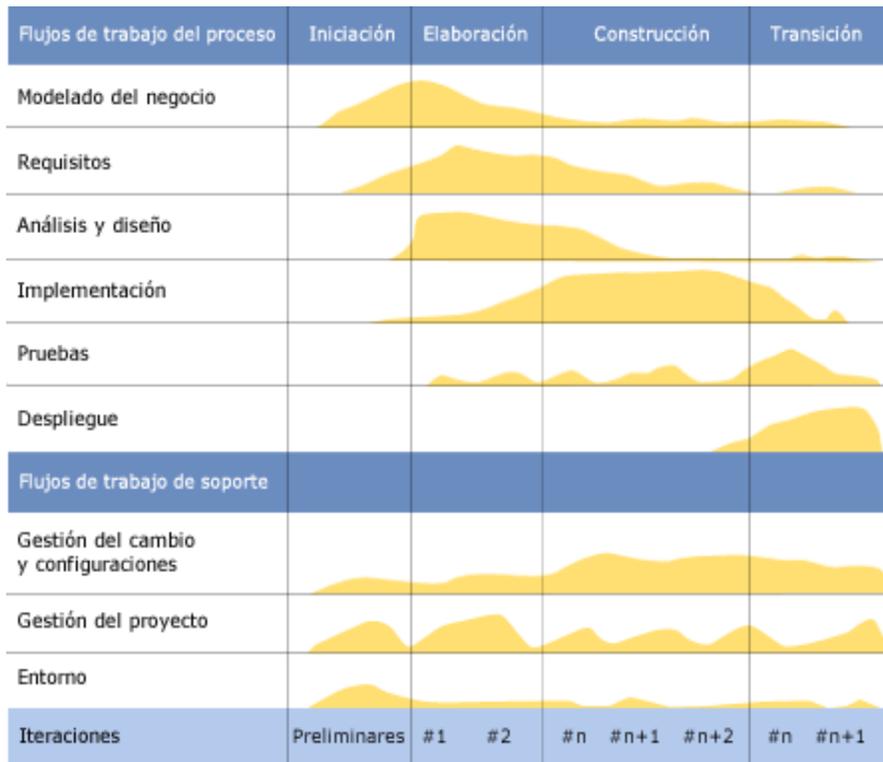
En la fase de elaboración, las iteraciones se orientan al desarrollo de la línea base de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la línea base de la arquitectura.

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

RUP comprende cuatro fases para el desarrollo de un proyecto: Inicio (También llamado Incepción), Elaboración, Desarrollo (Implementación, construcción) y la etapa de Cierre (Transición). Esta metodología agrupa las actividades en grupos lógicos definiéndose 9 flujos de trabajo de los cuales los 6 primeros son conocidos como flujos de ingeniería y los 3 últimos como flujos de apoyo, estos flujos de trabajo son: Modelado de Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba, Instalación, Gestión del cambio y configuraciones, Gestión del proyecto y Entorno.



**Figura 3** Ciclo de vida de RUP

### 3.3 Diferencias entre RUP y XP

El objetivo de establecer una comparación entre ambas metodologías no consiste en señalar cual es más eficiente sino describir las facilidades que brinda cada una para los diferentes tipos de proyectos.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

En el caso de proyectos muy grandes, RUP ha demostrado ser una metodología eficiente que junto a las herramientas que brinda el Rational se convierte en una excelente opción. El RUP no es una receta para hacer software, es una metodología muy seria acerca del proceso de creación del software y por tanto difícil de aprender. Esta metodología se caracteriza por su enfoque iterativo, cada iteración tiene entregables claros y en la medida de lo posible, el sistema debe ser funcional desde las primeras iteraciones de desarrollo sin embargo para proyectos pequeños o empresas que no cuentan con un gran departamento de desarrollo, el RUP es una alternativa costosa tanto en dinero como en tiempo. Con los años, han surgido alternativas a este enfoque protocolario sin descuidar la parte iterativa. Una rama de estas alternativas es el grupo de las metodologías ágiles entre las que se encuentra XP, que se enfocan más en el desarrollo que en su documentación es decir le dan más prioridad al desarrollo (al código y la funcionalidad) sin olvidar la documentación o la planeación.

La mayoría de los equipos que utilizan RUP lo han hecho por varios años y tienen especialistas del proceso que facilitan a los miembros sin experiencia el conocimiento de las prácticas y metodologías. La metodología XP es mucho más fácil de implementar y de aprender, por lo que los equipos jóvenes pueden incorporarla de manera más natural.

Los ciclos de vida de un proyecto en XP y en RUP no son exactamente iguales, aunque tienen bastantes similitudes, ambas son metodologías iterativas con probado éxito en el desarrollo de software.

Analizando las diferencias entre XP y RUP, se puede comprender que ambas metodologías parecen casi lo mismo, la diferencia está en el nivel de protocolo. En RUP el protocolo puede ser demasiado extenuante para un equipo pequeño y afectar directamente la productividad y velocidad del equipo. En XP no importa el tamaño del equipo, el énfasis está hecho en la comunicación dentro del equipo, incluyendo aquí por supuesto al cliente o usuario; en la velocidad del desarrollo y en la posibilidad de perfeccionar continuamente el código existente.

Concluyendo, la diferencia más importante entre la Programación Extrema y el Proceso Unificado del Rational es la agilidad en el desarrollo, conseguida mediante una comunicación intensiva del equipo y entre todos los desarrolladores, una disminución notable del protocolo y

de las jerarquías dentro y fuera del equipo y una autoevaluación intensiva incluso a nivel de cada desarrollador.

### **4. Lenguaje de Programación Web**

Con el comienzo de Internet y la programación Web, se desfasaron los diseños gráficos tradicionales, con lo que se empezaron a diseñar interfaces concretas para este medio, buscando ficheros pequeños para facilitar la carga de los mismos. La programación web se orientaba a un diseño muy cargado e interactuando con el usuario, mientras que al empezar a competir con millones de webs se ha optado más por el diseño sencillo y de fácil comprensión.

En algunas bibliografías el término lenguaje de programación está descrito como un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina. (4)

También es definido como una herramienta que nos permiten crear programas y software. (5).

El estudio de los conceptos anteriores permite concluir que un lenguaje de programación diseña el conjunto de acciones consecutivas a ejecutarse y que lleva implícito una organización semántica y sintáctica en función de obtener un resultado esperado.

Con la alta demanda que surgió a nivel mundial y debido a las tendencias y necesidades de las plataformas, comenzó a notarse una revolución en cuanto al nacimiento y desarrollo de nuevos lenguajes de programación web y aparece un nuevo profesional de la informática el diseñador web, considerado un experto en programación y cuyo papel laboral se define entre el diseñador gráfico tradicional y el programador de aplicaciones llevadas a Internet. (7)

#### **4.1 Lenguaje PHP**

Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Se caracteriza por su mayor meditación para cubrir las necesidades de las aplicaciones Web actuales gracias a la extensa librería de funciones con la que está dotado. (1)

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Es un lenguaje de programación utilizado para la creación de sitio web. PHP es un acrónimo recursivo que significa “PHP Hypertext Pre-processor”, (inicialmente se llamó Personal Home Page). Surgió en 1995, desarrollado por PHP Group.

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, similar al ASP de Microsoft o el JSP de Sun, embebido en páginas HTML y ejecutado en el servidor.

La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a Java Script o a C, para aquellos que conocen estos lenguajes. (8)

A diferencia de Java o Java Script que se ejecutan en el navegador PHP se ejecuta en el servidor por eso permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado es enviado al navegador. El resultado es normalmente una página HTML pero también podría ser una página WML.



## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

Figura 4 Funcionamiento del lenguaje PHP.

### Ventajas:

- ✓ Muy fácil de aprender.
- ✓ Se caracteriza por ser un lenguaje muy rápido.
- ✓ Soporta en cierta medida la orientación a objeto. Clases y herencia.
- ✓ Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- ✓ Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- ✓ Capacidad de expandir su potencial utilizando módulos.
- ✓ Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Incluye gran cantidad de funciones.
- ✓ No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

### Desventajas:

- ✓ Se necesita instalar un servidor web.
- ✓ Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- ✓ La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- ✓ La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.
- ✓ Dificulta la modularización.
- ✓ Dificulta la organización por capas de la aplicación.

PHP es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor web en forma de módulo o ejecutado como un binario CGI separado, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier cosa que sea ejecutada en un servidor web sea insegura por naturaleza.

PHP está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI que Perl o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación.

Las principales características de PHP son: su rapidez; su facilidad de aprendizaje; su soporte multiplataforma tanto de diversos Sistemas Operativos, como servidores HTTP y de bases de datos; y el hecho de que se distribuye de forma gratuita bajo una licencia abierta.

### 4.2 Java Script

Constituye el lenguaje del lado del cliente más utilizado por su alta compatibilidad con la mayoría de los navegadores además de permitir crear efectos especiales en las páginas y definir interactividades con los usuarios.

Java Script es un lenguaje con muchas posibilidades, facilita la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones y estructuras de datos complejas. Además, pone a disposición del programador todos los elementos que forman la página Web, para que éste pueda acceder a ellos y modificarlos dinámicamente.

Es un lenguaje altamente utilizado para validar los datos que son introducidos por el usuario a través de los formularios. Se hace necesario destacar que existen dos tipos de Java Script: por un lado está el que se ejecuta en el cliente, este es el Java Script propiamente dicho, aunque técnicamente se denomina Navigator JavaScript, y también existe un Javascript que se ejecuta en el servidor, es más reciente y se denomina LiveWire Javascript.

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

### 5. AJAX

Es una técnica de desarrollo Web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

AJAX es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

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

Se caracteriza por la recuperación asíncrona de datos ya que el usuario no tiene que esperar después de una petición, posee un acercamiento del ambiente de escritorio a la Web además de reducir el tamaño de la información intercambiada.

### **6.Wamp**

WAMP es el acrónimo usado para describir un sistema de infraestructura de internet. Provee a los desarrolladores con los cuatro elementos necesarios para un servidor web: un sistema operativo, un manejador de base de datos, un software para servidor web y un software de programación script.

El uso de un WAMP permite servir páginas html a internet, además de poder gestionar datos en ellas, al mismo tiempo un WAMP, proporciona lenguajes de programación para desarrollar aplicaciones Web.

Aunque los programas mencionados no fueron diseñados para trabajar conjuntamente entre ellos, su combinación es famosa por su bajo coste y por la compatibilidad de sus componentes Este conjunto de aplicaciones es una alternativa viable a los paquetes comerciales existentes y tiene una fácil instalación.

### 7. Sistemas de Gestión de Contenido

Un Sistema de gestión de contenidos (*Content Management System* en inglés, abreviado CMS) es un programa que permite crear una estructura de soporte (framework) para la creación y administración de contenidos por parte de los participantes principalmente en páginas web. Permite manejar de manera independiente el contenido por una parte y el diseño por otra.

Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido del sitio. El sistema permite manejar de manera independiente el contenido y el diseño siendo posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación en el sitio a varios editores.

Un sistema de administración de contenido a menudo funciona en el servidor del sitio web. Muchos sistemas proporcionan diferentes niveles de acceso dependiendo del usuario, variando si es el administrador, editor, o creador de contenido. El acceso al CMS se realiza generalmente a través del navegador web, y a veces se requiere el uso de FTP para subir contenido, normalmente fotografías o audio.

Los creadores de contenido publican sus documentos en el sistema. Los editores comentan, aceptan o rechazan los documentos. El editor en jefe es responsable por publicar el trabajo en el sitio. El CMS controla y ayuda a manejar cada paso de este proceso, incluyendo las labores técnicas de publicar los documentos a uno o más sitios. En muchos sitios con CMS una sola persona hace el papel de creador y editor.

Lo esencial de un trabajo con un gestor de contenido es que el que se halla seleccionado sea amigable y el administrador tenga absoluto control sobre el código que finalmente va a generarse.

Existen numerosos tipos de CMS. Se agrupan según el tipo de contenido que permiten gestionar en las aplicaciones, los CMS genéricos que son uno de los más conocidos y representativos, ofrecen la plataforma necesaria para desarrollar e implementar aplicaciones que den solución a necesidades específicas. Pueden servir para construir soluciones de gestión de contenidos.

### 7.1 Ejemplos de CMS

#### Joomla

Joomla es un sistema de administración de contenidos de código abierto construido con PHP bajo una licencia GPL. Este administrador de contenidos se usa para publicar en Internet e intranets utilizando una base de datos MySQL. En Joomla se incluyen características como: hacer caché de páginas para mejorar el rendimiento, indexamiento web, versiones imprimibles de páginas, flash con noticias, blogs, foros, encuestas, calendarios, búsqueda en el sitio web, e internacionalización del lenguaje.

#### MODx

Es un sistema de administración de contenidos y framework o CMF (Content Manager Framework) de código abierto, desarrollado en PHP. Su interfaz de programación de aplicaciones y modelo de eventos le otorga una amplia capacidad de adaptación y personalización haciéndolo muy flexible en la creación de sitios Web a diferencia de otros CMS rígidos basados en bloques. Su alto soporte de estándares HTML/CCS lo hace ideal para diseñadores Web.

MODx proporciona un framework para desarrollar sitios Web, Intranets y aplicaciones Web. Posee un completo soporte para el contenido (plano, HTML, XML), apariencia (CSS) y funcionalidad (PHP).

#### Drupal

Sistema gestor de contenido que permite publicar artículos, imágenes u otros archivos y servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios y permiso. Es un sistema dinámico: en lugar de almacenar sus contenidos en archivos estáticos en el sistema de ficheros del servidor de forma fija, el contenido textual de las páginas y otras configuraciones son almacenados en una base de datos y se editan utilizando un entorno Web incluido en el producto.

Drupal ha llegado a ser mucho más que sólo un portal de noticias gracias a su arquitectura flexible. Está constituido por una infraestructura base y un conjunto de módulos que ofrecen un

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

amplio conjunto de funciones. Drupal se usa por ejemplo en intranets de compañías, enseñanza en línea, comunidades de arte y administración de proyectos.

No solo es un software libre brindándoles a los usuarios la libertad sobre su producto adquirido pudiendo ser distribuido libremente, sino que es dotado por una gran modularidad como cualquier herramienta de este tipo, permite subir el contenido mediante formularios web, configurar roles de usuario diferentes, categorizar los contenidos además de brindar la oportunidad de que los visitantes pueden comentar las entradas. (9)

### **Gestión de usuarios**

Autenticación de usuarios: los usuarios se pueden registrar e iniciar sesión de forma local o utilizando un sistema de autenticación externo como Jabber, Blogger, LiveJournal u otro sitio Drupal. Para su uso en una intranet.

Permisos basados en roles: los administradores de Drupal no tienen que establecer permisos para cada usuario. En lugar de eso, pueden asignar permisos a un 'rol' y agrupar los usuarios por roles.

### **Gestión de contenido**

Control de versiones: el sistema de control de versiones de Drupal permite seguir y auditar totalmente las sucesivas actualizaciones del contenido: qué se ha cambiado, la hora y la fecha, quién lo ha cambiado, y más. También permite mantener comentarios sobre los sucesivos cambios o deshacer los cambios recuperando una versión anterior.

Enlaces permanentes (Permalinks): todo el contenido creado en Drupal tiene un enlace permanente asociado a él para que pueda ser enlazado externamente sin temor de que el enlace falle en el futuro.

Objetos de Contenido (Nodos): el contenido creado en Drupal es, funcionalmente, un objeto (Nodo). Esto permite un tratamiento uniforme de la información, como una misma cola de moderación para envíos de diferentes tipos, promocionar cualquiera de estos objetos a la página principal o permitir comentarios o no sobre cada objeto.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

Plantillas (Templetes): El sistema de temas de Drupal separa el contenido de la presentación permitiendo controlar o cambiar fácilmente el aspecto del sitio web. Se pueden crear plantillas con HTML y/o con PHP.

### **Blogging**

Agregador de noticias: incluye un potente Agregador de Noticias para leer y publicar enlaces a noticias de otros sitios web. Incorpora un sistema de cache en la base de datos, con temporización configurable.

Soporte de Blogger API: la interfaz de programación de aplicaciones (API) de Blogger permite que un sitio Drupal sea actualizado utilizando diversas herramientas, que pueden ser 'herramientas web' o 'herramientas de escritorio' que proporcionen un entorno de edición más manejable.

### **Plataforma**

Independencia de la base de datos: aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones. Drupal incorpora una capa de abstracción de base de datos que actualmente está implementada y mantenida para MySQL y PostgreSQL, aunque permite incorporar fácilmente soporte para otras bases de datos.

Multiplataforma: ha sido diseñado desde el principio para ser multi-plataforma. Puede funcionar con Apache o Microsoft IIS como servidor web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X. Por otro lado, al estar implementado en PHP, es totalmente portable.

Múltiples idiomas y Localización: está pensado para una audiencia internacional y proporciona opciones para crear un portal multilingüe. Todo el texto puede ser fácilmente traducido utilizando una interfaz web, importando traducciones existentes o integrando otras herramientas de traducción.

### **Administración y Análisis**

Administración vía Web: La administración y configuración del sistema se puede realizar enteramente con un navegador y no precisa de ningún software adicional.

Análisis, Seguimiento y Estadísticas: puede mostrar en las páginas web de administración informes sobre referrals (enlaces entrantes), popularidad del contenido, o de cómo los usuarios navegan por el sitio.

Registros e Informes: toda la actividad y los sucesos del sistema son capturados en un registro de eventos, que puede ser visualizado por un administrador.

### **Características de comunidad**

Comentarios enlazados: proporciona un potente modelo de comentarios enlazados que posibilita seguir y participar fácilmente en la discusión sobre el comentario publicado. Los comentarios son jerárquicos, como en un grupo de noticias o un foro.

Encuestas: incluye un módulo que permite a los administradores y/o usuarios crear encuestas on-line totalmente configurables.

Foros de discusión: incorpora foros de discusión para crear sitios comunitarios vivos y dinámicos.

Libro Colaborativo: esta característica es única de Drupal y permite crear un proyecto o "libro" a ser escrito y que otros usuarios contribuyan contenido. El contenido se organiza en páginas cómodamente navegables.

### **Rendimiento y escalabilidad**

Control de congestión: Drupal incorpora un mecanismo de control de congestión que permite habilitar y deshabilitar determinados módulos o bloques dependiendo de la carga del servidor. Este mecanismo es totalmente configurable y ajustable.

Sistema de Cache: El mecanismo de cache elimina consultas a la base de datos incrementando el rendimiento y reduciendo la carga del servidor.

### 8. Sistemas Gestores de base de datos

Los sistemas gestores o manejadores de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Facilita el proceso de definir, construir y manipular la base de datos (conjunto de datos) para diversas aplicaciones.

El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de los datos.

#### 8.1 PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional multihilo y multiusuario con más de seis millones de instalaciones además de ser orientada a objetos y ser un software libre. Se caracteriza por su amplia variedad de tipos nativos ya que provee nativamente soporte para números de precisión arbitraria, para texto de largo ilimitado, figuras geométricas y direcciones de IP. Presenta también una alta concurrencia pues permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos. (12)

### 8.2 MySQL

Sistema de gestión de bases de datos relacional. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

### 8.3 Microsoft SQL Server

Constituye un sistema manejador de datos capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, Sybase ASE, PostgreSQL, Interbase, Firebird o MySQL.

La potencia de este sistema es evaluado a través de sus características más destacables entre las cuales podemos mencionar su soporte de transiciones, la escalabilidad, estabilidad y seguridad que brinda. Soporta procedimientos almacenados y permite trabajar cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.

Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET, pero el servidor sólo está disponible para Sistemas Operativos Windows. (14)

## 9. Propuesta para el desarrollo

En el capítulo se realizó un estudio sobre los diferentes sistemas gestores de base de datos y de contenido y se investigó sobre dos ejemplos de metodologías de desarrollo, en donde se enfatizó en una metodología ágil y en otra de tipo robusta para establecer comparaciones entre

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

---

ambas y lograr llegar a una decisión en función de cual utilizar. Además se expusieron las arquitecturas a emplear en la implementación de la aplicación web.

A partir de todo lo expuesto en el desarrollo de este capítulo se propone utilizar como metodología para guiar todo el proceso de investigación la metodología XP por ser fácil de aplicar a un proyecto de pocos desarrolladores y por buscar la simplicidad y la retroalimentación. Como sistema gestor de base de datos PostgreSQL ya que es un manejador de datos de tipo libre. Entre los sistema gestores de contenido se tomó como decisión hacer uso de Drupal por ser un sistema dinámico con una arquitectura flexible, es de tipo software libre y está dotado por un conjunto de módulos además de seleccionarse un paquete de desarrollo como lo constituye Wamp pues esta compuesto por diferentes programas que unidos se transforman en un poderosa herramienta organizativa para el proceso de implementación que se anhela llevar a cabo. Se decidió hacer uso de la arquitectura de tres capas así como de la arquitectura cliente-servidor, la primera en función de lograr una organización física y la segunda como línea organizativa que construya la lógica de la aplicación que se desea llevar a cabo. El lenguaje de programación elegido es PHP como exigencia de los requisitos que debe tener la aplicación satisfaciendo las exigencias del cliente y con el propósito de lograr la validación del sistema se hace uso del lenguaje del lado del cliente Java Script apoyado en la potente técnica Ajax contribuyendo con la seguridad de la aplicación.

### **10. Conclusiones**

En este capítulo se caracterizaron las aplicaciones Web incluyendo sus consideraciones técnicas. Se expusieron las diferentes tecnologías para desarrollar una aplicación de este tipo haciendo énfasis en las posibles metodologías que podían ser usadas, los sistemas manejadores de datos siendo más explícitos en el CMS Drupal que es a nivel mundial uno de los preferidos. Además en el desarrollo del capítulo se analizaron varios sistemas gestores de base de datos con el propósito de seleccionar el que más se adaptara a las exigencias de la aplicación.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

---

### CAPITULO 2. CARACTERÍSTICAS DEL SISTEMA

#### **2.1 Introducción**

En este capítulo se persigue como objetivo hacer una explicación de las características principales del sistema que se pretende implementar, prestando atención principal al problema que dio lugar a la investigación que se ha llevado a cabo en función de darle resultado a la situación problemática planteada así como hacer énfasis en temas tan importantes como lo constituye el objeto de automatización, la propuesta del sistema, el flujo de trabajo del mismo y las personas relacionadas a la aplicación en cuestión.

#### **2.2 Objeto de automatización.**

Como la Universidad de las Ciencias Informáticas en la actualidad no cuenta con una aplicación que permita difundir la vida y obra revolucionaria de Julio Antonio Mella, se hace necesaria la implementación de un sistema que gestione la información necesaria y suficiente para lograr alcanzar la divulgación que se desea por parte los clientes.

Para llevar a cabo los intereses ya planteados es de gran importancia efectuar la automatización de algunos procesos tales como la creación de cuentas de usuarios así como la publicación de materiales y la descarga de los mismos. Son procesos que se hacen vitales para obtener un resultado final óptimo y deseado.

#### **2.3 Propuesta del sistema.**

El concepto sistema ha sido definido de varias maneras. Algunos autores como L. von Bertalanffy lo describen de la manera siguiente: "Un sistema es un conjunto de unidades en interrelación". Según Ferdinand de Saussure: "Sistema es una totalidad organizada, hecha de elementos solidarios que no pueden ser definidos más que los unos con relación a los otros en función de su lugar en esa totalidad". Para el IEEE Standard Dictionary of Electrical and Electronic Terms sistema es un todo integrado, aunque compuesto de estructuras diversas, interactuantes y especializadas. Cualquier sistema tiene un número de objetivos, y los pesos asignados a cada uno de ellos pueden variar ampliamente de un sistema a otro. Un sistema

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

---

ejecuta una función imposible de realizar por una cualquiera de las partes individuales. La complejidad de la combinación está implícita. (6)

El sistema es un todo que no puede ser tomado en partes sin que se pierdan sus características esenciales y, por lo tanto, se debe estudiar como un todo. Consiste en subunidades con ciertas condiciones de frontera entre las cuales se dan procesos de transporte. (7)

Basándose en cada definición dada y de acuerdo a los estudios realizados el sistema constituye un conjunto unitario organizado que está compuesto por varias partes y que todas tienen una función fundamental dentro del mismo.

Un proyecto que se pretende trabajar correctamente debe estar formado como un potente sistema que labore en conjunto. El trabajo propone un sistema automatizado que brinde ciertas funcionalidades que sean de comodidad para aquellos usuarios que interactúen con el mismo. Entre las actividades a desarrollarse dentro de la aplicación se encuentran la creación de cuentas a usuarios que deseen intercambiar y navegar dentro del sistema, brinda la posibilidad de mostrar el perfil de usuario así como modificarlo. Entre otras de las funcionalidades que se desean automatizar se encuentran la publicación de materiales (ya sean videos, imágenes y textos) y la descarga de las publicaciones realizadas al igual que el empleo de un foro que facilite la comunicación entre usuarios autenticados, abordando el tema rector de la aplicación. En el sistema habrá habilitada una encuesta que permitirá a los administradores comprobar el nivel de utilidad que presenta la aplicación para los usuarios facilitando a los administradores el trabajo en función de satisfacer las necesidades de los miembros de la aplicación.

En el caso de los usuarios que sean visitantes, estos podrán navegar dentro de la aplicación consultando los documentos que estarán habilitados pero no tendrán autorización a la descarga de los materiales, al intercambio dentro del foro y poder votar dentro de la encuesta ya que son facilidades que brindara la aplicación únicamente a aquellos usuarios autenticados, además de brindársele la oportunidad de crear una cuenta. De manera general las facilidades del sistema se mostraran de acuerdo al nivel de privilegio que tengan las personas que interactúen con este.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

---

El sistema será una aplicación Web que utilizara el gestor de contenidos Drupal y cada una de las funcionalidades se implementaran como parte de los diferentes módulos que estarán integrados dentro del gestor y que tendrán un propósito específico.

### **2.3 Flujo del sistema.**

El sistema se pretende que funcione como una herramienta de divulgación. Su flujo de desarrollo comienza cuando el visitante interactúa con la aplicación donde puede obtener información referente al tema que se aborda. Si el usuario que visita desea crearse una cuenta de usuario, puede llenar los campos del perfil que desea tener dentro del sistema. Para poder intercambiar con las diferentes secciones que hay habilitadas, el usuario debe autenticarse antes y se comprueba que sea miembro para tener acceso pleno a cada una de las funcionalidades de importancia, en caso contrario solamente tiene acceso a la navegación y obtención de información de manera más restringida. Cuando una persona pasa a ser miembro del sistema se le envía un mensaje de confirmación con la contraseña y el usuario expuesto en el perfil.

Si se desea eliminar una cuenta de usuario, se envía un mensaje de solicitud y este debe ser llegado al administrador siendo el autorizado para eliminar dicha cuenta y quitarle privilegios al usuario solicitante enviando posteriormente un mensaje de confirmación. Al concluir el proceso de creación de cuentas de usuario y el de autenticación, el usuario deja de ser visitante y pasa a ser usuario autenticado

### **2.4 Personas relacionadas con la aplicación.**

Se le denomina en el trabajo persona relacionada con el sistema a quien interactúe con el sistema y que obtenga resultados deseados a partir del proceso de desarrollo de alguna de las funcionalidades que presenta la aplicación así como toda aquella que intercambie con la misma pero que solamente tengo acceso a la navegación sin poder hacer uso de las secciones privilegiadas del sistema.

## CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA

Tabla 1 Personal asociado a la aplicación.

Personas relacionadas con la aplicación	Justificación de la participación
Visitante	Constituye la persona que navega dentro de la aplicación Web sin haberse creado una cuenta que le permita poder tener acceso a los servicios que brinda la aplicación. Esta persona puede familiarizarse con el entorno en el cual navega y posteriormente crearse una cuenta de usuario que le posibilite disfrutar de las facilidades que proporciona la aplicación.
Usuario Autenticado o Miembro	Es la persona que ya ha sido autenticada dentro de la aplicación y que disfruta de los servicios que brinda la misma a los usuarios que tiene una cuenta creada. Tiene pleno acceso al servicio de descarga de diferentes documentos así como de videos e imágenes, puede hacer uso de las numerosas bibliografías con que cuenta la aplicación , puede debatir dentro del foro y responder encuestas online.
Administrador	Es la persona encargada de llevar a cabo la administración de todo el contenido que se maneja dentro de la aplicación al igual que las cuentas de los usuarios que han sido autenticados. Es de manera general la persona facultada para la gestión del sistema.

### **2.5 Conclusiones**

En este capítulo se han expuesto las principales características que debe tener el sistema en función de dar una propuesta de desarrollo de la implementación de la aplicación que se desea automatizar. A favor de dicha propuesta se expusieron los principales procesos a automatizar así como aquellas personas que se relacionaran con el sistema de acuerdo al rol que desempeñen dentro del mismo.

A partir de todo lo manifestado a lo largo del capítulo se permite comenzar con la próxima fase de desarrollo del sistema en función del cumplimiento de los requerimientos impuestos por el cliente para obtener un resultado posterior confiable y satisfactorio.

## CAPÍTULO 3. EXPLORACIÓN Y PLANIFICACIÓN

### CAPITULO 3. EXPLORACIÓN Y PLANIFICACIÓN

#### 3.1 Introducción

En este capítulo se hace referencia a todo lo relacionado con la fase de exploración y planificación que aconseja la metodología seleccionada por el equipo de desarrolladores del sistema.

#### 3.2 Fase exploración.

En la etapa de exploración el cliente plantea a grandes rasgos las historias de usuario y al mismo tiempo el equipo de desarrollo se familiariza con la tecnología, herramientas y prácticas que se utilizarán en el proyecto. Se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo.

##### 3.2.1 Historias de usuarios.

Las historias de usuario se utilizan con el mismo propósito que los casos de uso lo que presentando algunas diferencias como que constan de tres o cuatro líneas escritas por el cliente sin hacer énfasis en los detalles y expresadas en un lenguaje no técnico. Se emplean para estimar tiempo de desarrollo de la parte de la aplicación que describen. El tratamiento de las historias de usuario es muy dinámico y flexible. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas ya que cada una es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

Durante el proceso de exploración se identificaron 13 historias de usuarios que serán descritas a continuación:

Tabla 2 HU Crear Cuenta de Usuario.

Numero: 1	Nombre: Crear cuenta de usuario.
Usuario: Visitante	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Bajo
Puntos Estimación: 1	Iteración Asignada: 1

## CAPÍTULO 3. EXPLORACIÓN Y PLANIFICACIÓN

<p>Descripción: El usuario cuando entra a la aplicación se le brinda la oportunidad de crearse una cuenta para que posteriormente pueda tener uso pleno de los diferentes servicios que se brinda dentro de la aplicación y poder tener acceso a sus diferentes secciones.</p>
<p>Observaciones:</p>

**Tabla 3 HU Mostrar Perfil de Usuario.**

Historia de Usuario	
Numero: 2	Nombre: Mostrar perfil de usuario.
Usuario: Usuario Autenticado	
Prioridad en Negocio: Media	Riesgo en desarrollo: Bajo
Puntos Estimación: 1	Iteración Asignada: 2
<p>Descripción: Brinda la posibilidad de ver los datos que el usuario lleno en los diferentes campos necesarios al efectuar la creación de la cuenta para accesibilidad dentro de la aplicación.</p>	
<p>Observaciones: Para que un usuario pueda tener vista a su perfil debe antes haber creado una cuenta.</p>	

**Tabla 4 HU Modificar Perfil de Usuario.**

Historia de Usuario	
Numero: 3	Nombre: Modificar perfil de usuario
Usuario: Usuario Autenticado	
Prioridad en Negocio: Media	Riesgo en desarrollo: Bajo
Puntos Estimación: 1	Iteración Asignada: 2
<p>Descripción: Brinda al usuario la posibilidad de modificar algún dato expuesto en la creación de</p>	

## CAPÍTULO 3. EXPLORACIÓN Y PLANIFICACIÓN

su cuenta.
Observaciones: Para que un usuario pueda tener acceso a la modificación de su perfil antes debe tener una cuenta creada.

**Tabla 5 HU Autenticar Usuario.**

Historia de Usuario	
Numero: 4	Nombre: Autenticar usuario
Usuario: Usuario Autenticado	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Medio
Puntos Estimación: 1	Iteración Asignada: 1
Descripción: Posibilita que la persona que acceda al sistema introduzca sus datos (usuario y contraseña) para verificar que tiene una cuenta creada y poder darle acceso a los servicios que brinda la aplicación.	
Observaciones:	

**Tabla 6 HU Eliminar Cuenta de Usuario.**

Historia de Usuario	
Numero: 5	Nombre: Eliminar cuenta de usuario
Usuario: Administrador	
Prioridad en Negocio: Media	Riesgo en desarrollo: Bajo
Puntos Estimación: 1	Iteración Asignada: 2
Descripción: El administrador se encarga de eliminar una cuenta de usuario que ya no tenga validez dentro de la aplicación.	

## CAPÍTULO 3. EXPLORACIÓN Y PLANIFICACIÓN

Observaciones: Para eliminar una cuenta antes se debe verificar que este creada.

**Tabla 7 HU Publicar Materiales.**

Historia de Usuario	
Numero: 6	Nombre: Publicar materiales
Usuario: Administrador	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Alto
Puntos Estimación: 2	Iteración Asignada: 1
Descripción: El administrador es el encargado de publicar dentro de la aplicación aquellos materiales informativos y audiovisuales que son utilizados por los usuarios.	
Observaciones:	

**Tabla 8 HU Eliminar Materiales.**

Historia de Usuario	
Numero: 7	Nombre: Eliminar materiales
Usuario: Visitante	
Prioridad en Negocio: Media	Riesgo en desarrollo: Bajo
Puntos Estimación: 1	Iteración Asignada: 2
Descripción: El administrador es el encargado de eliminar los materiales que se consideren ya no son necesarios dentro de la aplicación y que no cumplan un objetivo importante en la misma.	
Observaciones:	

## CAPÍTULO 3. EXPLORACIÓN Y PLANIFICACIÓN

**Tabla 9 HU Descargar Publicaciones.**

Historia de Usuario	
Numero: 8	Nombre: Descargar publicaciones
Usuario: Usuario autenticado	
Prioridad en Negocio: Alta	Riesgo en desarrollo: Medio
Puntos Estimación: 1	Iteración Asignada: 2
Descripción: Permite a los usuarios que estén autenticados realizar la descarga de las publicaciones.	
Observaciones:	

**Tabla 10 HU Gestionar Noticias.**

Historia de Usuario	
Numero: 9	Nombre: Gestionar noticias
Usuario: Administrador	
Prioridad en Negocio: Media	Riesgo en desarrollo: Bajo
Puntos Estimación: 1	Iteración Asignada: 3
Descripción: Incluye las acciones de publicar y eliminar noticias. El administrador es el encargado de publicar las noticias de carácter nacional e internacional que sean relevantes así como podrá eliminar aquellas que ya no estén actualizadas.	
Observaciones:	

**Tabla 11 HU Buscar Información.**

Historia de Usuario

## CAPÍTULO 3. EXPLORACIÓN Y PLANIFICACIÓN

Numero: 10	Nombre: Buscar información
Usuario: Visitante	
Prioridad en Negocio: Baja	Riesgo en desarrollo: Bajo
Puntos Estimación: 1	Iteración Asignada: 3
Descripción: Proporciona a cualquier usuario que interactúe con el sistema como un invitado, obtener información.	
Observaciones:	

**Tabla 12 HU Gestionar Foro.**

Historia de Usuario	
Numero: 11	Nombre: Gestionar foro
Usuario: Administrador	
Prioridad en Negocio: Media	Riesgo en desarrollo: Medio
Puntos Estimación: 1	Iteración Asignada: 2
Descripción: Incluye las acciones de responder mensajes expuestos en el foro y borrar mensajes que ya han sido respondidos dentro del foro.	
Observaciones:	

**Tabla 13 HU Gestionar Encuesta.**

Historia de Usuario	
Numero: 12	Nombre: Gestionar encuesta
Usuario: Administrador	

## CAPÍTULO 3. EXPLORACIÓN Y PLANIFICACIÓN

Prioridad en Negocio: Baja	Riesgo en desarrollo: Bajo
Puntos Estimación: 1	Iteración Asignada: 3
Descripción: Incluye las acciones de publicar pregunta y eliminar. El administrador recoge los resultados de la encuesta y la almacena para determinar el grado de aceptación de la aplicación. Después de un tiempo determinado elimina la pregunta y publica otra.	
Observaciones:	

**Tabla 14 Mostrar información.**

Historia de Usuario	
Numero: 13	Nombre: Mostrar información
Usuario: Administrador	
Prioridad en Negocio: Media	Riesgo en desarrollo: Bajo
Puntos Estimación: 1	Iteración Asignada: 2
Descripción: Permite mostrar información importante al administrador como los usuarios que mas visitan la aplicación así como las secciones de mayor gusto para los miembros del sistema. .	
Observaciones:	

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

## CAPÍTULO 3. EXPLORACIÓN Y PLANIFICACIÓN

### 3.3 Fase de Planificación.

Dentro de la etapa de planificación una de las principales tareas de desarrollo esta expresadas en priorizar las historias de usuario que constituyen el primer paso en cualquier proyecto que siga la metodología XP como guía para su crecimiento, además en esta fase se lleva a cabo una estimación del esfuerzo con el cual debe contar el grupo de desarrolladores para efectuar cada historia de usuario planteada así como la prioridad con que serán implementadas. Se expresa esta estimación utilizando como medida el punto que constituye una semana de trabajo ideal donde se labora el tiempo planeado sin interrupción.

La etapa de planificación permite determinar la velocidad del proyecto pues facilita el conteo de las historias de usuarios que se pueden efectuar en una iteración, es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Aconseja la programación en pareja incrementando la productividad y la calidad del producto planificado al igual que las reuniones diarias donde se expongan los problemas y soluciones de forma conjunta entre el equipo de trabajo, en general es una etapa en la cual se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.

Tabla 15 Estimación de esfuerzo por HU.

Historia de usuario	Puntos de Estimación
Crear cuenta usuario	1
Mostrar perfil de usuario	1
Modificar perfil de usuario	1
Autenticar usuario	1
Eliminar cuenta de usuario	1
Publicar materiales	2
Eliminar materiales	1

## CAPÍTULO 3. EXPLORACIÓN Y PLANIFICACIÓN

Descargar publicaciones	1
Gestionar noticias	1
Buscar información	1
Gestionar foro	1
Gestionar encuesta	1
Mostrar información	1

### 3.3.1 Plan de iteraciones

Las iteraciones dentro de un proyecto constituyen la llave para obtener una retroalimentación sobre el desarrollo y evolución del mismo. Un Plan de Iteración tiene un fuerte peso por constituir una especie de herramienta para los desarrolladores brindando un mecanismo honesto en el cual se expresa con precisión un tiempo de desarrollo del producto. De manera general el Plan de Iteración recoge la planificación detallada de un periodo corto de tiempo dentro del proyecto.

Dentro de la fase de planificación se identificaron tres iteraciones para efectuar la implementación del sistema y las cuales son detalladas a continuación:

#### Iteración #1

En esta iteración es donde se decide establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto y tiene como objetivo la implementación de las historias de usuarios que tengan una prioridad alta dentro del negocio y que sean elevadamente importantes en la primera etapa de implementación. Se hace referencia a historias de usuarios como: Crear cuenta de usuario, Autenticar usuario, Publicar materiales. Se tendrá la primera versión de prueba correspondiente a esta iteración para mostrar al cliente y obtener una opinión del mismo persiguiendo como propósito una retroalimentación para el equipo de desarrolladores del sistema.

## CAPÍTULO 3. EXPLORACIÓN Y PLANIFICACIÓN

### Iteración #2

En esta iteración se tiene como meta principal la implementación de aquellas historias de usuarios que tiene una prioridad alta pero que no son implementadas en la primera iteración ejemplo la Historia de usuario Descargar publicaciones así como aquellas funcionalidades de prioridad media las cuales están dadas en las Historias de usuarios: Mostrar perfil de usuario, Modificar perfil de usuario, Eliminar cuenta de usuario, Eliminar materiales, Gestionar noticias, Gestionar foro y Mostrar información. En esta iteración la prueba correspondiente a la misma se hace para mostrar al cliente y partir de este punto realizar algún cambio de acuerdo a la opinión dada. Es en este paso en donde se hace una valoración del nivel de satisfacción del cliente con respecto al sistema hasta la iteración en la cual se laboró.

### Iteración #3

El objetivo fundamental de la última iteración esta dado por la implementación de aquellas historias de usuarios que tiene una prioridad muy baja dentro del negocio entre las cuales se encuentran Buscar información y Gestionar encuesta, además de permitir corregir errores anteriores en las iteraciones que ya pasaron. Como resultado visible de esta iteración se encuentra la obtención de la primera versión final del producto la cual se pondrá en funcionamiento para comprobar su desempeño y rendimiento óptimo en función de satisfacer las expectativas del cliente.

### 3.3.2 Plan de duraciones

Se hace necesario crear el plan de duraciones de las iteraciones con el objetivo fundamental de manifestar el tiempo de demora de cada una de las iteraciones y el orden en que serán implementadas las historias de usuarios en cada una de estas iteraciones.

**Tabla 16 Plan de duración de iteraciones para el equipo de trabajo.**

Iteración	Orden de las Historias de Usuario a implementar	Duración Total de la Interacción
Iteración #1	Crear cuenta de usuario. Autenticar usuario.	4 semanas

## CAPÍTULO 3. EXPLORACIÓN Y PLANIFICACIÓN

	Publicar materiales.	
Iteración #2	<p>Descargar publicaciones.</p> <p>Mostrar perfil de usuario.</p> <p>Modificar perfil de usuario</p> <p>Eliminar cuenta de usuario.</p> <p>Eliminar materiales.</p> <p>Gestionar noticias.</p> <p>Gestionar foro.</p> <p>Mostrar información</p>	8 semanas
Iteración #3	<p>Buscar información.</p> <p>Gestionar encuesta.</p>	2 semanas

**Tabla 17 Módulos e Historias de Usuarios Abarcadas.**

Módulos	Historias de usuarios que abarca
Usuario	<p>1-Crear cuenta de usuario.</p> <p>2-Autenticar usuario.</p> <p>3-Buscar información.</p> <p>4- Descargar publicaciones</p>
Administrador	<p>1-Eliminar cuenta de usuario.</p> <p>2- Gestionar encuesta.</p> <p>3-Gestionar foro.</p>

## CAPÍTULO 3. EXPLORACIÓN Y PLANIFICACIÓN

Material	1-Publicar materiales. 2-Eliminar materiales.
Perfil	1-Mostrar perfil de usuario 2-Modificar perfil de usuario.
Noticias	1-Gestionar noticias.
Listado	1-Mostrar información

**Tabla 18 Plan de duración de entrega.**

Módulos	Final primera iteración 2da Semana de Febrero	Final segunda iteración 3ra Semana de Abril	Final tercera iteración 3ra Semana de Mayo
Usuario	0.1	1.0	Finalizado
Administrador		0.1	1.0
Material	0.1	1.0	Finalizado
Perfil		1.0	Finalizado
Noticias		1.0	Finalizado
Listado			1.0

### 3.4 Conclusiones

Con la realización de este capítulo se ha llevado a cabo todo lo referente a la etapa de exploración y planificación que lleva implícita la metodología XP. Se expusieron las historias de usuario que caracterizan al sistema así como una estimación de la duración de cada una. Además, a lo largo del capítulo se explicaron las iteraciones que tendrá la implementación de la aplicación, los módulos que ha consideración del equipo de desarrollo estarán presentes y el plan de duración de entrega del sistema.

#### 4.1 Introducción

La metodología XP constituye una guía de implementación lo suficientemente ágil y sencilla sin dejar de llevar implícita una calidad significativa en el producto final. Para lograr los resultados deseados se debe hacer un análisis detallado del diseño, el desarrollo así como de las pruebas a aplicar al sistema que se desea automatizar siguiendo las reglas de dicha metodología.

En el presente capítulo se explican aquellas características con que contara el diseño de la aplicación, se detallan las iteraciones realizadas durante la etapa de codificación del proyecto además de exponer aquellas tareas aplicadas a cada historia de usuario y las pruebas de aceptación que se deben manifestar sobre el sistema siempre intentando obtener un producto con los requerimientos exigidos.

#### 4.2 Diseño

La metodología XP hace un énfasis significativo en aquellos diseños que se caracterizan por su simplicidad y su alto nivel de claridad permitiendo un trabajo con una evolución constante y un desarrollo fácil para el equipo que labora en función de lograr obtener un producto.

La metodología Extreme Programming propone utilizar el término nuevo metáfora en su fase de diseño definiendo una historia de cómo funciona el sistema completo, sugiriendo este concepto como una manera sencilla de explicar el propósito del proyecto y poder guiar la estructura y arquitectura del mismo. Se hace necesario que tanto el cliente como el equipo de desarrolladores comparta la metáfora para lograr que todos los implicados dentro del desarrollo del sistema conversen el mismo idioma y puedan dialogar en función del crecimiento del producto, además una buena metáfora permitirá tener suficiente contenido que sirva de guía a la arquitectura del proyecto.

La recodificación constituye otra propuesta que brinda la metodología seleccionada basándose en escribir nuevamente parte del código de un programa sin cambiar sus funcionalidades y con el único propósito de hacerlo más simple y entendible. Llevando a cabo la refactorización dentro de un proyecto estamos logrando el reciclaje del código manteniéndolo limpio y fácil de

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

comprender, modificar y ampliar permitiendo dentro del ciclo de vida de un proyecto ahorrar en tiempo e incrementado su calidad.

Para diseñar la aplicación se deberá cumplir con tres principios fundamentales Cargo o Clase, Responsabilidad y Colaboración, estos términos también son conocidos dentro de la metodología como las tarjetas CRC, utilizando esta técnica para lograr un desprendimiento del método de trabajo basado en procedimientos y laborar sobre una metodología que se basa en objetos. Estas tarjetas brindan la posibilidad que el equipo completo de desarrolladores contribuya en la tarea de diseño.

La tarjeta CRC se caracteriza por representar un objeto. El nombre de la clase se coloca en forma de título en la tarjeta, en la parte izquierda las funcionalidades y las clases que se implican en cada funcionalidad se ponen en la derecha.

Tabla 19 Tarjeta CRC del Módulo Usuario

Modulo Usuario	
Funcionalidades	Colaboraciones
Crear cuenta	Block
Autenticar usuario	Menu
Buscar información	Search
Descargar publicaciones	

Tabla 20 Tarjeta CRC del Módulo Administrador

Modulo Administrador	
Funcionalidades	Colaboraciones
Eliminar cuenta de usuario.	Usuario
Publicar pregunta encuesta	Menu
Eliminar pregunta encuesta	Block

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

Publicar mensajes del foro	Administrador
Responder mensajes del foro	Node
Eliminar mensajes del foro	Comment

**Tabla 21 Tarjeta CRC del Módulo Material.**

Modulo Usuario	
Funcionalidades	Colaboraciones
Publicar materiales	Block
Eliminar materiales	Menu Administrador

**Tabla 22 Tarjeta CRC del Módulo Perfil**

Modulo Perfil	
Funcionalidades	Colaboraciones
Mostrar perfil	Usuario
Modificar perfil	Block Menu Administrador

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

Tabla 23 Tarjeta CRC del Módulo Noticia

Modulo Noticia	
Funcionalidades	Colaboraciones
Publicar noticias	Administrador
Eliminar noticias	Menu
	Block
	Node

Tabla 24 Tarjeta CRC al Módulo Listado

Modulo Listado	
Funcionalidades	Colaboraciones(Módulos)
Mostrar información	Usuario
	Block
	Menu
	Administrador

### 4.3 Desarrollo.

La fase de desarrollo de un sistema constituye uno de los pilares fundamentales para obtener un producto final deseado con una óptima calidad. Está basado en principios específicos que se han de poner en práctica y los cuales están dados en la disponibilidad del cliente no solo para ayudar al equipo de desarrolladores sino formando parte del mismo pues una de las condiciones principales de la metodología está dada en la constante comunicación con el cliente logrando en cada momento la retroalimentación que hace factible la evolución del sistema. Otro principio consistente de esta fase lo constituye la estandarización del código permitiendo facilitar la lectura y la modificación del mismo por parte de cualquier miembro del equipo.

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

La metodología XP en esta fase también propone la programación en pareja donde las personas involucradas trabajaran de forma conjunta frente a un ordenador. Es una estrategia para minimizar los errores y a la vez lograr mejores diseños.

La fuerza de esta fase está dada además en la programación dirigida por las pruebas. La metodología XP propone un proceso inverso a las metodologías tradicionales que realizan las pruebas después de implementado el producto. En función de lo planteado se crean las pruebas antes que el código para posteriormente llevar a cabo la implementación del mismo permitiendo que esta actividad sea más rápida y resulte beneficiosa en aquellos diseños de sistemas complicados. Estas pruebas son nombradas pruebas unitarias realizadas por los desarrolladores y la definición de estas pruebas al comienzo condiciona o dirigen el desarrollo.

Durante la fase de planificación se expusieron las iteraciones que a consideración del grupo de trabajo iba a llevar la implementación del sistema a automatizar.

Teniendo en cuenta esta planificación se mencionan las iteraciones generadas por la misma así como las tareas a desarrollar para cada Historia de usuario haciendo responsable un grupo de desarrolladores o una persona.

### 4.3.1 Iteración 1

En esta primera iteración se desarrollan las Historias de Usuario de prioridad alta dentro del sistema donde su implementación brinda una versión del producto a mostrar al cliente con las funcionalidades críticas y que son esencialmente necesarias.

Tabla 25 Módulos Abordados en la Primera Iteración.

Módulos	Historias de usuario	Tiempo de Implementación	
		Estimación	Real
Usuario	Crear Cuenta de Usuario.	1	0.5
	Autenticar Usuario.	1	0.5
Material	Publicar Materiales.	2	2

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

En función de una mejor evolución del sistema en su etapa de desarrollo se llevan a cabo algunas tareas las cuales serán expresadas a continuación y que comprenden las historias de usuario a implementar en la primera iteración.

### Módulo Usuario

Tabla 26 Tarea 1 al Módulo Usuario

Tarea	
Numero de tarea: 1	Numero HU: 1
Nombre de la tarea: Configuración del modulo usuario	
Tipo de tarea: Configuración	Puntos Estimados: 1
Fecha inicio: 9 de febrero de 2009	Fecha fin: 13 de febrero de 2009
Programador responsable: Ariam Ajete Mena.	
Descripción: Se especifican los datos que serán de utilidad en la creación de una cuenta de usuario así como el rol a la cual estará sujeta la cuenta. Estos datos posteriormente se almacenaran en la base de datos.	

Tabla 27 Tarea 2 al Módulo Usuario.

Tarea	
Numero de tarea: 2	Numero HU:4
Nombre de la tarea: Autenticar Usuario	
Tipo de tarea: Configuración	Puntos Estimados: 0.5
Fecha inicio: 13 de febrero de 2009	Fecha fin: 16 de febrero de 2009
Programador responsable: Ariam Ajete-María de los Ángeles Santos	
Descripción: Se gestiona el proceso de verificar los datos introducidos por el usuario con	

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

los almacenados en la base de datos y se le asigna el rol correspondiente dentro del sistema.

### Módulo Material

Tabla 28 Tarea 1 al Módulo Material.

Tarea	
Numero de tarea: 1	Numero HU: 6
Nombre de la tarea: Creación del módulo material.	
Tipo de tarea: Configuración	Puntos Estimados: 1
Fecha inicio: 10 de febrero de 2009	Fecha fin: 17 de febrero de 2009
Programador responsable: María de los Ángeles Santos-Ariam Ajete Mena	
Descripción: Se activan los archivos propios del módulo <i>upload</i> que vienen predefinidos en el código del gestor de contenido seleccionado.	

Tabla 29 Tarea 2 al Módulo Material

Tarea	
Numero de tarea: 2	Numero HU: 6,7
Nombre de la tarea: Publicación de materiales	
Tipo de tarea: Configuración	Puntos Estimados: 0.14
Fecha inicio: 17 de febrero de 2009	Fecha fin: 24 de febrero de 2009
Programador responsable: María de los Ángeles Santos	
Descripción: Se publican todos los materiales que lleva la aplicación.	

### 4.3.2 Iteración 2

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

En esta iteración se efectúa la elaboración de los módulos administrador, listado, perfil, noticia así como se continúa con la implementación de los módulos material y usuario.

**Tabla 30 Módulos Abordados en la Segunda Iteración**

Módulos	Historias de usuario	Tiempo de Implementación	
		Estimación	Real
Administrador	Eliminar Cuenta de Usuario	1	0.5
	Gestionar Foro	1	0.5
Material	Eliminar Materiales.	1	0.5
Usuario	Descargar Publicaciones	1	1
Perfil	Mostrar Perfil de Usuario.	1	0.5
	Modificar Perfil de Usuario.	1	0.5
Noticia	Gestionar Noticia.	1	1
Listado	Mostrar Información	1	1

### Módulo Administrador

**Tabla 31 Tarea 1 al Modulo Administrador.**

Tarea	
Numero de tarea:1	Numero HU: 5
Nombre de la tarea: Eliminar cuenta de usuario.	

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

Tipo de tarea: Configuración.	Puntos Estimados: 0.5
Fecha inicio: 3 de marzo de 2009	Fecha fin: 8 de marzo de 2009
Programador responsable: Ariam Ajete.	
Descripción: El administrador es el encargado de eliminar la cuenta del usuario en la base de datos.	

**Tabla 32 Tarea 2 al Modulo Administrador**

Tarea	
Numero de tarea: 2	Numero HU: 11
Nombre de la tarea: Creación del módulo foro.	
Tipo de tarea: Configuración	Puntos Estimados: 0.5
Fecha inicio: 12 de marzo de 2009	Fecha fin: 16 de marzo de 2009
Programador responsable: Ariam Ajete	
Descripción: Se configura el modulo <i>forum</i> usando el gestor de contenido seleccionado que brinda las líneas de comando predeterminadas para este módulo.	

### Módulo Material

**Tabla 33 Tarea 3 al Modulo Material**

Tarea	
Numero de tarea: 3	Numero HU: 7
Nombre de la tarea: Eliminación de materiales.	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 12 de marzo del 2009	Fecha fin: 16 de marzo de 2009

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

Programador responsable: María de los Ángeles Santos
Descripción: El administrador accede a los materiales publicados en el sistema y los elimina según considere necesario.

### Módulo Usuario

Tabla 34 Tarea 3 al Módulo Usuario.

Tarea	
Numero de tarea: 3	Numero HU: 8
Nombre de la tarea: Descarga de publicaciones.	
Tipo de tarea: Configuración	Puntos Estimados: 1
Fecha inicio: 16 de marzo del 2009	Fecha fin: 23 de marzo de 2009
Programador responsable: Ariam Ajete.	
Descripción: Se descargan las publicaciones que fueron expuestas con el uso del módulo <i>upload</i> y que facilita la descarga ya que el código predefinido del mismo brinda esa posibilidad.	

### Módulo Perfil

Tabla 35 Tarea 1 al Modulo Perfil

Tarea	
Numero de tarea:1	Numero HU: 2,3
Nombre de la tarea: Configuración del módulo perfil	
Tipo de tarea: Configuración	Puntos Estimados: 0.5
Fecha inicio: 24 de marzo de 2009	Fecha fin: 29 de abril de 2009

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

Programador responsable: Ariam Ajete
Descripción: Se activan los archivos propios del módulo <i>profile</i> para ser posteriormente visualizados y llevar a cabo las acciones de mostrar el perfil y modificar perfil.

### Módulo Noticia

Tabla 36 Tarea 1 al Módulo Noticia

Tarea	
Numero de tarea:1	Numero HU:9
Nombre de la tarea: Creación del código fuente del módulo noticia.	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.5
Fecha inicio: 23 de marzo de 2009	Fecha fin: 27 de marzo de 2009
Programador responsable: María de los Ángeles Santos	
Descripción: Se desarrolla el código que facilitara las funcionalidades que presentara el módulo dentro del sistema. Se utilizan diferentes <i>hook</i> . Se crean los archivos <i>.info</i> , <i>.install</i> y <i>.module</i>	

Tabla 37 Tarea 2 al Módulo Noticia

Tarea	
Numero de tarea: 2	Numero HU: 9
Nombre de la tarea: Creación de la tabla en la base de datos perteneciente al módulo noticia.	
Tipo de tarea: Configuración	Puntos Estimados: 0.14
Fecha inicio: 27 de marzo de 2009	Fecha fin: 28 de abril de 2009

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

Programador responsable: María de los Ángeles Santos
Descripción: Se crea una tabla dentro de la base de datos con los datos necesarios de almacenamiento de una noticia.

**Tabla 38 Tarea 3 al Módulo Noticia**

Tarea	
Numero de tarea: 3	Numero HU:9
Nombre de la tarea: Configuración del módulo noticia	
Tipo de tarea: Configuración	Puntos Estimados: 0.28
Fecha inicio: 29 de abril de 2009	Fecha fin: 31 de marzo de 2009
Programador responsable: María de los Ángeles Santos-Ariam Ajete.	
Descripción: Se incorpora a la interfaz gráfica del sistema después de ser configurado.	

### Modulo Listado

**Tabla 39 Tarea 1 al Módulo Listado**

Tarea	
Numero de tarea:1	Numero HU: 13
Nombre de la tarea: Creación del código fuente del módulo listado	
Tipo de tarea: Desarrollo	Puntos Estimados: 0.71
Fecha inicio: 2 de abril de 2009	Fecha fin: 7 de abril de 2009
Programador responsable: María de los Ángeles Santos-Ariam Ajete.	
Descripción: Se implementa el código que facilitara las funcionalidades a presentar el módulo. Se utilizan diferentes <i>hook</i> . Se crean los archivos <i>.info</i> , <i>.install</i> , <i>.module</i> .	

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

**Tabla 40 Tarea 2 al Módulo Listado**

Tarea	
Numero de tarea:2	Numero HU: 13
Nombre de la tarea: Configuración del módulo listado	
Tipo de tarea: Configuración	Puntos Estimados: 0.28
Fecha inicio: 7 de abril de 2009	Fecha fin: 9 de abril de 2009
Programador responsable: María de los Ángeles Santos-Ariam Ajete.	
Descripción: Se configura el módulo para incorporarlo a la interfaz del sistema.	

### 4.3.3 Iteración 3

En esta iteración se implementaron aquellas funcionalidades de baja prioridad. Se le da continuación al modulo administrador al igual que el modulo usuario. Al finalizar esta iteración se pretende obtener un producto final esperado que cumpla con los requerimientos del cliente.

**Tabla 41 Módulos Abordados en la Tercera Iteración.**

Módulos	Historias de usuario	Tiempo de Implementación	
		Estimado	Real
Usuario	Buscar Información.	1	0.2
Administrador	Gestionar Encuesta.	1	0.5

### Módulo Usuario

**Tabla 42 Tarea 3 al Módulo Usuario**

Tarea
-------

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

Numero de tarea:4	Numero HU:10
Nombre de la tarea: Buscar información	
Tipo de tarea: Configuración	Puntos Estimados:
Fecha inicio: 12 de abril de 2009	Fecha fin: 14 de abril del 2009
Programador responsable: Ariam Ajete.	
Descripción: Se activa el módulo <i>search</i> propio del gestor de contenido seleccionado que brinda las líneas de código predeterminadas para este módulo.	

### Módulo Administrador

Tabla 43 Tarea 3 al Módulo Administrador

Tarea	
Numero de tarea:3	Numero HU:12
Nombre de la tarea: Configuración del modulo encuesta.	
Tipo de tarea: Configuración	Puntos Estimados: 0.42
Fecha inicio: 18 de abril del 2009	Fecha fin: 21 de abril de 2009
Programador responsable: María de los Ángeles Santos	
Descripción: Se activa el módulo <i>poll</i> propio del gestor de contenido seleccionado que proporciona las líneas de código predeterminadas para la creación de este módulo.	

### 4.4 Pruebas

Las pruebas como parte de las tareas a realizar por un grupo de desarrolladores de un sistema que se rige por la metodología XP constituyen una de los pilares básicos. Las pruebas están

integradas en el proceso de crecimiento y evolución de un proyecto además de brindar la posibilidad de permitir el excelente resultado posterior que se desea alcanzar y convertirse en una potente herramienta de desarrollo.

La metodología XP propone dos tipos de pruebas:

- Pruebas Unitarias.
- Pruebas de Aceptación

### 4.3.1 Pruebas Unitarias.

Las pruebas unitarias deben ser definidas antes de realizarse la implementación del código. Todos los módulos deben de pasar por las pruebas antes de ser publicados en caso contrario se considerara que el código no está completo. Es de gran importancia que el conjunto de pruebas sea guardado junto con el código para que pueda ser utilizado por otros desarrolladores en caso necesario, es decir en caso de tener que cambiar, corregir o recodificar el código.

### 4.3.2 Pruebas de Aceptación

Las pruebas de aceptación son creadas a partir de las historias de usuario permitiendo determinar que cada historia de usuario ha sido implementada satisfactoriamente al final de la iteración donde esté involucrada. Este importante y necesario periodo de prueba se conoce como periodo de caja negra y cada prueba de este tipo que se realice representa una salida esperada del sistema. Una historia de usuario puede tener todas las pruebas de aceptación que sean necesarias para lograr determinar su completo funcionamiento. Una historia de usuario no puede ser considerada terminada hasta que no ha pasado por sus pruebas de aceptación.

Los clientes son responsables de verificar que los resultados de las pruebas sean correctos para lograr alcanzar la garantía de que todos los requerimientos han sido cumplidos y el sistema es aceptable, además están en la obligación de que en caso de fallar varias pruebas indicar el orden de prioridad de solución a los problemas que se manifiestan.

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

Ya que todas las tareas realizadas en cada fase de la metodología XP se hacen a través de un trabajo en colectivo y la responsabilidad es grupal, es recomendable la publicación de los resultados obtenidos a partir de la realización de las pruebas de aceptación con el objetivo de lograr que todo el equipo de desarrolladores esté al tanto de la información que se brinda con la implementación de dichas pruebas.

Tablas de Pruebas de Aceptación a los Diferentes Módulos.

**Tabla 44 Prueba 1 al Módulo Administrador**

Caso de Prueba de Aceptación	
Código: HU5_P1	HU: 5
Nombre: Eliminar Cuenta de Usuario.	
Descripción: Prueba para la funcionalidad de eliminar una cuenta de usuario.	
Condiciones de Ejecución:  El usuario debe tener una cuenta ya creada.  Envía mensaje de solicitud de eliminación de cuenta al administrador.	
Entradas/Pasos de Ejecución:  El usuario que desea se le elimine una cuenta envía mensaje de solicitud al administrador, este lo recibe, busca el usuario en la base de datos y lo elimina.	
Resultado Esperado:  El usuario es eliminado de la base de datos sin provocar errores.	
Evaluación de la Prueba:  Prueba Satisfactoria.	

**Tabla 45 Prueba 2 al Módulo Administrador**

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

Caso de Prueba de Aceptación	
Código: HU11_P1	HU:11
Nombre: Responder mensajes.	
Descripción: Prueba para la funcionalidad de responder mensajes dentro del foro.	
Condiciones de Ejecución:  Para poder interactuar dentro el foro los usuarios deben haber sido autenticados con anterioridad.	
Entradas/Pasos de Ejecución:  El usuario se autentica ya sea como miembro o como administrador y accede al foro donde puede responder a los mensajes de otros usuarios.	
Resultado Esperado:  Los usuarios que interactúen dentro del foro pueden responder mensajes dejados por otros usuarios.	
Evaluación de la Prueba:  Prueba Satisfactoria.	

Tabla 46 Prueba 3 al Módulo Administrador

Caso de Prueba de Aceptación	
Código: HU11_P2	HU:11
Nombre: Eliminar mensajes.	
Descripción: Prueba para la funcionalidad de eliminar mensajes dentro del foro.	
Condiciones de Ejecución:	

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

<p>Para poder eliminar mensajes publicados en el foro el usuario debe autenticarse y tener el rol de administrador.</p>
<p>Entradas/Pasos de Ejecución:</p> <p>El usuario autenticado debe ser administrador.</p> <p>Para eliminar algún mensaje publicado en el foro debe haber sido respondido.</p>
<p>Resultado Esperado:</p> <p>El administrador elimina los mensajes satisfactoriamente.</p>
<p>Evaluación de la Prueba:</p> <p>Prueba Satisfactoria.</p>

**Tabla 47 Prueba 4 al Módulo Administrador**

Caso de Prueba de Aceptación	
Código: HU_12P1	HU:12
Nombre: Publicar pregunta de encuesta	
Descripción: Prueba para la funcionalidad de publicar pregunta de la encuesta.	
Condiciones de Ejecución:	
El administrador es el encargado de publicar la pregunta de la encuesta.	
Entradas/Pasos de Ejecución:	
El usuario se autentica como administrador para poder publicar la pregunta de la encuesta.	
Resultado Esperado:	
La pregunta es publicada sin errores dentro del sistema.	

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

Evaluación de la Prueba:  Prueba Satisfactoria.
---

**Tabla 48 Prueba 5 al Módulo Administrador**

Caso de Prueba de Aceptación	
Código: HU12_P2	HU:12
Nombre: Eliminar pregunta.	
Descripción: Prueba para la funcionalidad de eliminar pregunta de la encuesta.	
Condiciones de Ejecución:  La pregunta debe haber sido publicada.  El administrador es el encargado de eliminar la pregunta de la encuesta.	
Entradas/Pasos de Ejecución:  El usuario se autentica como administrador para poder eliminar la pregunta de la encuesta.	
Resultado Esperado:  La pregunta es eliminada de la encuesta sin errores dentro del sistema.	
Evaluación de la Prueba:  Prueba Satisfactoria.	

**Tabla 49 Prueba 1 al Módulo Usuario**

Caso de Prueba de Aceptación	
Código: HU10_P1	HU:10
Nombre: Buscar Información.	

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

Descripción: Prueba para la funcionalidad de la búsqueda de información.
Condiciones de Ejecución:  El usuario no debe ser necesariamente un miembro del sistema.
Entradas/Pasos de Ejecución:  El usuario que desea buscar información es solamente un visitante, no tiene que autenticarse y puede acceder a la información sin descargar algún material.
Resultado Esperado:  El usuario puede obtener información dentro del sistema.
Evaluación de la Prueba:  Prueba Satisfactoria.

**Tabla 50 Prueba 2 al Módulo Usuario**

Caso de Prueba de Aceptación	
Código: HU1_P1	HU:1
Nombre: Crear Cuenta de Usuario	
Descripción: Prueba para la funcionalidad de crear una cuenta al usuario	
Condiciones de Ejecución:	
Entradas/Pasos de Ejecución:  El visitante tiene la oportunidad de crearse una cuenta llenado su perfil.	
Resultado Esperado:  El usuario llena su perfil, se le crea una cuenta dentro del sistema y podrá acceder	

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

posteriormente a las funcionalidades propias del sistema para los usuarios que son miembros.

Evaluación de la Prueba:

Prueba Satisfactoria.

**Tabla 51 Prueba 3 al Modulo Usuario**

Caso de Prueba de Aceptación	
Código: HU4_P1	HU:4
Nombre: Autenticar usuario	
Descripción: Prueba para la funcionalidad de autenticar usuario	
Condiciones de Ejecución: El usuario debe tener una cuenta creada para poder ser autenticado.	
Entradas/Pasos de Ejecución: El usuario entra la contraseña y se verifica dentro del sistema que es miembro del mismo. En caso afirmativo puede acceder a todas las secciones de la aplicación.	
Resultado Esperado: El usuario es autenticado satisfactoriamente e intercambia con el sistema.	
Evaluación de la Prueba: Prueba Satisfactoria.	

**Tabla 52 Prueba 4 al Módulo Usuario**

Caso de Prueba de Aceptación
------------------------------

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

Código: HU8_P1	HU:8
Nombre: Descargar Materiales.	
Descripción: Prueba para la funcionalidad de descargar materiales.	
<p>Condiciones de Ejecución:</p> <p>El usuario debe ser autenticado dentro del sistema para poder descargar.</p> <p>Los materiales deben haber sido publicados con anterioridad.</p>	
<p>Entradas/Pasos de Ejecución:</p> <p>El usuario se autentica dentro del sistema y puede acceder a la descarga de materiales ya sean imágenes, videos y/o documentos.</p>	
<p>Resultado Esperado:</p> <p>El usuario autenticado o miembro del sistema puede efectuar la descarga de materiales sin dificultades.</p>	
<p>Evaluación de la Prueba:</p> <p>Prueba Satisfactoria.</p>	

**Tabla 53 Prueba 1 al Módulo Material**

Caso de Prueba de Aceptación	
Código: HU6_P1	HU:6
Nombre: Publicar Materiales.	
Descripción: Prueba para la funcionalidad de la publicación de materiales.	
<p>Condiciones de Ejecución:</p> <p>El usuario que se autentica tiene que ser administrador.</p>	

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

<p>Entradas/Pasos de Ejecución:</p> <p>Los materiales deben ser publicados dentro de la aplicación por el administrador del sistema.</p>
<p>Resultado Esperado:</p> <p>La publicación de los materiales se realiza sin errores.</p>
<p>Evaluación de la Prueba:</p> <p>Prueba Satisfactoria.</p>

**Tabla 54 Prueba 2 al Módulo Material**

Caso de Prueba de Aceptación	
Código: HU7_P1	HU:7
Nombre: Eliminar Materiales.	
Descripción: Prueba para la funcionalidad de eliminar materiales.	
<p>Condiciones de Ejecución:</p> <p>El usuario que se autentica debe ser administrador.</p> <p>Los materiales a eliminar deben haber sido publicados con anterioridad.</p>	
<p>Entradas/Pasos de Ejecución:</p> <p>Se eliminan los materiales por parte del administrador.</p>	
<p>Resultado Esperado:</p> <p>La eliminación de los materiales se realiza sin errores.</p>	
<p>Evaluación de la Prueba:</p>	

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

Prueba Satisfactoria.

**Tabla 55 Prueba 1 al Módulo Perfil**

Caso de Prueba de Aceptación	
Código: HU2_P1	HU:2
Nombre: Mostrar Perfil de Usuario.	
Descripción: Prueba para la funcionalidad de mostrar el perfil de un usuario.	
Condiciones de Ejecución: El usuario debe tener una cuenta ya creada donde lleno su perfil.	
Entradas/Pasos de Ejecución: El usuario que desea ver su perfil selecciona dentro de la aplicación la opción de ir al perfil.	
Resultado Esperado: El usuario accede a su perfil sin provocar errores dentro del sistema.	
Evaluación de la Prueba: Prueba Satisfactoria.	

**Tabla 56 Prueba 2 al Módulo Perfil**

Caso de Prueba de Aceptación	
Código: HU3_P1	HU:3
Nombre: Modificar Perfil	
Descripción: Prueba para la funcionalidad de modificar el perfil de un usuario.	

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

<p>Condiciones de Ejecución:</p> <p>El usuario debe tener una cuenta ya creada.</p> <p>Para modificar su perfil debe acceder a este primero.</p>
<p>Entradas/Pasos de Ejecución:</p> <p>El usuario que desea modificar su perfil primero accede a la vista del mismo y posteriormente puede cambiar algún campo deseado.</p>
<p>Resultado Esperado:</p> <p>El usuario modifica su perfil sin errores dentro del sistema.</p>
<p>Evaluación de la Prueba:</p> <p>Prueba Satisfactoria.</p>

**Tabla 57 Prueba 1 al Modulo Noticias**

<b>Caso de Prueba de Aceptación</b>	
Código: HU9_P1	HU:9
Nombre: Gestionar Noticias.	
Descripción: Prueba para la funcionalidad de publicar noticias	
<p>Condiciones de Ejecución:</p> <p>El administrador es el encargado de publicar las noticias.</p>	
<p>Entradas/Pasos de Ejecución:</p> <p>El administrador publica las noticias nacionales e internacionales seleccionadas.</p>	
<p>Resultado Esperado:</p> <p>El administrador publica las noticias sin dificultades y los usuarios pueden acceder a las</p>	

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

mismas.
Evaluación de la Prueba:  Prueba Satisfactoria.

**Tabla 58 Prueba 2 al Módulo Noticias**

<b>Caso de Prueba de Aceptación</b>	
Código: HU11_P2	HU:11
Nombre: Eliminar Noticias.	
Descripción: Prueba para la funcionalidad de eliminar noticias.	
Condiciones de Ejecución:  El administrador es el encargado de eliminar las noticias.  Para poder eliminar noticias ya deben haber sido publicadas.	
Entradas/Pasos de Ejecución:  El administrador entra a la sección de las noticias y elimina las noticias consideradas desactualizadas.	
Resultado Esperado:  El administrador elimina las noticias del sistema sin dificultades.	
Evaluación de la Prueba:  Prueba Satisfactoria.	

**Tabla 59 Prueba 1 al Módulo Listado**

<b>Caso de Prueba de Aceptación</b>
-------------------------------------

## CAPÍTULO 4. DISEÑO, DESARROLLO Y PRUEBA

Código: HU13_P2	HU:13
Nombre: Mostrar Información	
Descripción: Prueba para la funcionalidad de mostrar información a los administradores	
Condiciones de Ejecución:  El administrador accede a la aplicación.  Ejecuta la acción deseada para obtener la información acerca de los usuarios y las secciones más visitadas del sistema.	
Entradas/Pasos de Ejecución:  El administrador entra al sistema con el objetivo de identificar las secciones más visitadas y los usuarios que frecuentan con mayor interés la aplicación.	
Resultado Esperado:  El administrador obtiene la información solicitada.	
Evaluación de la Prueba:  Prueba Satisfactoria.	

### 4.5 Conclusiones

En este capítulo se plantearon las tareas correspondientes para dar solución a las historias de usuario planteadas así como las pruebas de aceptación efectuadas con el objetivo de darle al cliente un producto que cumpla con las condiciones planteadas por el mismo. Con la conclusión de este capítulo se considera terminada la propuesta que brinda el trabajo.

## CONCLUSIONES GENERALES

---

### CONCLUSIONES GENERALES

En el proceso de desarrollo de la investigación llevada a cabo se efectuaron estudios de las tendencias actuales que tienen las aplicaciones Web y su nivel de aceptación mundialmente con el propósito de determinar el grado de aprobación por parte de aquellos que tuvieron interacción con el sistema a automatizar, además de manifestar la necesidad de crear una aplicación que divulgara la labor de una figura de la historia cubana. Con la realización del trabajo se expone un sistema automatizado que propone hacer uso de numerosos materiales que facilita al usuario una fuente de conocimientos al igual que otras secciones que elevan la calidad del sistema en cuestión, dando solución a los requerimientos exigidos por parte del cliente.

Para lograr la ampliación del trabajo se empleó como guía del proceso la metodología XP que facilitó la organización de la labor efectuada y permitir una comunicación factible entre los desarrolladores y el cliente; logrando obtener los resultados anhelados en el software. Se logro definir, diseñar e implementar el modelo a seguir por parte del equipo de desarrolladores posibilitando obtener una aplicación web con los servicios específicos que se ansiaba presentara.

Se considera concluido el trabajo con el cumplimiento de todos los objetivos emitidos al inicio de la investigación suministrando como resultado final un producto que cumpla con las pretensiones expuestas por el cliente.

## RECOMENDACIONES

---

### RECOMENDACIONES

Como resultado del proceso de desarrollo efectuado se recomienda:

- 1- La utilización del presente trabajo como bibliografía para futuras investigaciones que utilicen la misma línea de implementación expuesta.
- 2- El empleo del sistema para personas que pretendan indagar sobre la vida revolucionaria de Julio Antonio Mella.
- 3- Efectuar la implementación de Aplicaciones Web que expongan la labor histórica de figuras cubanas.

## REFERENCIAS BIBLIOGRÁFICAS

---

## REFERENCIAS BIBLIOGRÁFICAS

1. **Alvarez, Miguel Angel.** DesarrolloWeb. [En línea] [Citado el: 12 de febrero de 2009.] <http://www.desarrolloweb.com>.
2. **Sanchez, Maria A. Mendoza.** informatizate. [En línea] 7 de junio de 2004. [Citado el: 14 de febrero de 2009.] <http://www.informatizate.net>.
3. **Escribano, Gerardo Fernandez.** Introducción a xtreme Programming. [En línea] 2002. [Citado el: 14 de febrero de 2009.] [www.info-ab.uclm.es](http://www.info-ab.uclm.es).
4. **Wesler, Rahlp.** Kioskea. [En línea] 16 de octubre de 2008. [Citado el: 16 de febrero de 2009.] <http://es.kioskea.net>.
5. **Jonson, Mark.** Lenguajes de Programación. [En línea] 2009. [Citado el: 16 de febrero de 2009.] [www.lenguajes-de-programacion.com](http://www.lenguajes-de-programacion.com).
6. **J.M.Raus.** Daedalus. [En línea] [Citado el: 17 de febrero de 2009.] <http://www.daedalus.es>.
7. **Valderrama, Angela Maria.** Aprendeonline. [En línea] 10 de mayo de 2006. [Citado el: 17 de febrero de 2009.] <http://aprendeonline.udea.edu.co>.
8. **Alvarez, Miguel Angel.** DesarrolloWeb. [En línea] [Citado el: 16 de febrero de 2009.] Miguel Angel Alvarez . <http://www.desarrolloweb.com>.
9. **Raus, J.M.** Daedalus. [En línea] [Citado el: 17 de febrero de 2009.] <http://www.daedalus.es>.

1. **SlideShare.** slideshare. *www.slideshare.net*. [En línea] desconocida, desconocido de desconocido de 2008. [Citado el: 25 de Enero de 2009.] <http://www.slideshare.net/Decimo/arquitectura-3-capas>.
2. **Cornejo, José Enrique González.** docirs. *www.docirs.cl*. [En línea] 21 de Diciembre de 2008. [Citado el: 2009 de Enero de 2009.] <http://www.slideshare.net/Decimo/arquitectura-3-capas>.
3. [En línea] [Citado el: 25 de Enero de 2009.] <http://www.proactiva-calidad.com/java/patrones/mvc.html>.
4. *wigahluk.wordpress.com* [En línea] Programming, Software Development, XP, 14 de Junio de 2007. [Citado el: 25 de Enero de 2009.] <http://wigahluk.wordpress.com/2007/06/14/percepciones-acerca-de-la-programacion-agil-y-xp/>.
5. *wigahluk.wordpress.com*. [En línea] Programming, RUP, Software Development, XP, 26 de Junio de 2007. [Citado el: 25 de Enero de 2009.] <http://wigahluk.wordpress.com/2007/06/26/entre-la-xp-y-el-rup/>.
6. **Bagüés, Ramiro Lago.** *www.proactiva-calidad.com*. [En línea] Proactiva. [Citado el: 25 de Enero de 2009.] <http://www.proactiva-calidad.com/java/patrones/mvc.html>.
7. **Valdés, Damián Pérez.** *www.maestrosdelweb.com*. [En línea] 2 de Noviembre de 2007. [Citado el: 25 de Enero de 2009.] <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>
8. *linuxcentro*. [En línea] 22 de Febrero de 2007. [Citado el: 25 de Enero de 2009.] <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.
9. **Reyero, J. A.** (16 de Marzo de 2006). Recuperado el 25 de Enero de 2009, de drupal.org.es: <http://drupal.org.es/drupal>.
10. (s.f.). Recuperado el 25 de Enero de 2009, de Rincondelvago: [http://html.rincondelvago.com/sistemas-gestores-de-bases-de-datos\\_1.html](http://html.rincondelvago.com/sistemas-gestores-de-bases-de-datos_1.html)

---

11. Vera, I. A. (16 de Marzo de 2004). Recuperado el 25 de Enero de 2009, de Biblioteca Virtual de Salud: [http://bvs.sld.cu/revistas/aci/vol12\\_2\\_04/aci06204..htm](http://bvs.sld.cu/revistas/aci/vol12_2_04/aci06204..htm)

12. (s.f.). Recuperado el 25 de Enero de 2009, de TLDP-ES: <http://es.tldp.org/Postgresql-es/web/navegable/tutorial/intro.html#AEN34>

13. (19 de Julio de 2002). Recuperado el 25 de Enero de 2009, de Desarrolloweb: <http://www.desarrolloweb.com/articulos/840.php>

14. (s.f.). Recuperado el 25 de Enero de 2009, de Microsoft: <http://www.microsoft.com/spain/sql/productinfo/overview/what-is-sql-server.msp>

### GLOSARIO DE TÉRMINOS

**Sistema Operativo:** Serie de programas, dispuestos ya sea en el software o en la memoria fija que hacen al hardware utilizable.

**HTML:** Siglas de **HyperText Markup Language** (*Lenguaje de Marcas de Hipertexto*), es el lenguaje de marcado predominante para la construcción de páginas Web.

**CSS:** Significa hojas de estilo en cascada y se utilizan y son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML.

**DOM:** Significa **Modelo en Objetos para la Representación de Documentos** y es esencialmente una interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos.

**Navegador Web:** Programa que permite visualizar la información que contiene una página web.

**Applet:** Componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador web.

**Adobe Flash Player:** Aplicación en forma de reproductor multimedia creado inicialmente por Macromedia y actualmente distribuido por Adobe Systems. Permite reproducir archivos SWF.

**Software:** Programas y datos almacenados en un ordenador.

**Hardware:** Substrato físico en el cual existe el software. Abarca todas las piezas físicas de un ordenador (CPU, placa base, etc.).

**Lenguaje de programación:** conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina.

**CGI:** Interfaz de Entrada Común (por sus siglas en inglés 'Common Gateway Interface') es de las primeras formas de programación Web dinámica. Es un método para la transmisión de información hacia un compilador instalado en el servidor. Su función principal es la de añadir una mayor interacción a los documentos web que por medio del HTML se presentan de forma estática.

**Framework:** Estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y

---

un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

**Asíncrona:** suceso que no tiene lugar en total correspondencia temporal con otro suceso.