PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS

DISERTACION PREVIA A LA OBTENCION DE TITULO DE
INGENIERO EN SISTEMAS

DESARROLLO DE UN SISTEMA DE ADMINISTRACIÓN DE TAREAS DE SOPORTE Y MANTENIMIENTO INFORMÁTICO

PAOLO ALEJANDRO AGUILAR PÉREZ

DIRECTOR: ING. JAVIER CONDOR

QUITO, 2011

AGRADECIMIENTO

El culminar una etapa de mi vida y empezar otra me llena de mucha satisfacción, por lo que la concepción de este proyecto está dedicada a mis padres, y a todas las personas que en el transcurso del estudio de mi carrera fueron pilares fundamentales en mi crecimiento profesional. Sin ellos, jamás hubiese podido conseguir lo que hasta ahora. Su tenacidad y lucha insaciable han hecho de ellos el gran ejemplo a seguir y destacar.

Índice

Índice de figuras

CAPITULO I

CAPITULO III

CAPITULO IV

Índice de tablas

CAPITULO III

Resumen

El presente proyecto de disertación de grado trata del Desarrollo de un Sistema de Administración de Tareas de Soporte y Mantenimiento Informático. El mismo que será construido siguiendo la metodología que plantea *Extreme Programming*, dado que ésta es destacada dentro del ámbito de los procesos ágiles de desarrollo de software.

Este sistema está enfocado a cualquier tipo de empresa que desee gestionar la asignación de tareas para el control de los servicios de soporte y mantenimiento informático.

Se encuentra desarrollada sobre la plataforma Microsoft .Net y JavaScript. Por lo que se requiere de una Base de datos SQL Server 2000+ y un servidor web con Internet Information Server (IIS) para poder visualizar los archivos con extensión ASPX. Estos requerimientos únicamente son a nivel de Servidor, dado que esta aplicación fue realizada con el objeto de que funcione en cualquier *browser* de internet.

Finalmente, esta disertación se encuentra dividida en cinco capítulos los cuales abarcan el marco teórico, el diseño, el análisis, la construcción y retroalimentación del sistema. Mediante los cuales se indica el procedimiento que se realizó para construir la aplicación anteriormente mencionada, y como debe ser llevado a cabo el proceso de desarrollo de software aplicando una metodología ágil.

1. CAPITULO I: Marco Teórico

En este capítulo se tratarán conceptos de ingeniería de software, Extreme Programming como metodología de desarrollo de software y nociones básicas de qué son y cómo se despliegan las aplicaciones sobre internet. Extreme Programming guiará la construcción del sistema planteado en esta disertación, con la mentalidad de que todo proyecto debe seguir un proceso que permita evaluar los resultados a la conclusión del mismo; y procurando colocar más énfasis en la adaptabilidad que en la previsibilidad.

1.1. Ingeniería de software

La ***ingeniería*** es el conjunto de conocimientos y técnicas científicas aplicadas a la invención y perfeccionamiento de todos sus diversos aspectos incluyendo la resolución y optimización de problemas que afectan directamente a los seres humanos en su actividad cotidiana. En la cual el conocimiento, manejo y dominio de las matemáticas y física, obtenido mediante estudio, experiencia y práctica, se aplica con juicio para desarrollar formas eficientes de utilizar los materiales y las fuerzas de la naturaleza para beneficio de la humanidad y del ambiente. [C]

Pese a que la ingeniería como tal está ligada al ser humano, su nacimiento como campo de conocimiento específico viene con el comienzo de la revolución industrial, constituyendo uno de los actuales pilares en el desarrollo de las sociedades modernas.

Otro concepto que define a la ingeniería es "el saber aplicar los conocimientos científicos a la invención, perfeccionamiento o utilización de la técnica en todas sus determinaciones. Esta aplicación se caracteriza por utilizar principalmente el ingenio de una manera más pragmática y ágil que el método científico, puesto que una actividad de ingeniería, por lo general, está limitada a un tiempo y recursos dados por proyectos. El ingenio implica tener una combinación de sabiduría e inspiración para modelar cualquier sistema en la práctica". [B]

Y, según la IEEE; el software es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación. Entre las cuales destacan: Métodos y metodologías, Procesos de desarrollo, Gestión de proyectos, Medición y estimación, Ingeniería de requisitos / requerimientos, Gestión de riesgos, usabilidad, evaluación, Métricas, Calidad, e Ingeniería Web. [1]

Por lo tanto se define a la ingeniería de software como, el grupo de métodos, técnicas y herramientas que se utilizan en la producción del software, más allá de la actividad principal de programación. La misma que mediante estudio, experiencia y práctica, se encarga de plasmar las ideas para dar soluciones ordenadas, estructuradas y acordes a la realidad.

Dado que la definición para Ingeniería descrita anteriormente es una percepción personal, citaré algunas definiciones que esta rama ha tenido en años anteriores.

- Ingeniería del Software es el estudio de los principios y metodologías para desarrollo y mantenimiento de sistemas de software. [Zelkovitz, 1978]

- Ingeniería del Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar y operar (funcionar) y mantenerlos. Así como también desarrollo de software o producción de software. [Bohem, 1976]
- La Ingeniería del Software es el establecimiento y uso de principios sólidos de la ingeniería para obtener económicamente un software confiable y que funcione de modo eficiente en máquinas reales. [Bauer, 1972]
- Ingeniería de Software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación (funcionamiento) y mantenimiento del software: es decir, la aplicación de ingeniería al software. [IEEE, 1993]
- La Ingeniería de Software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema hasta el mantenimiento de este después que se utiliza. [Sommerville, 2004]
- La Ingeniería de Software es una disciplina que integra el proceso, los métodos, y las herramientas para el desarrollo de software de computadora. [Pressman, 2005]

1.2. Metodología de desarrollo: Extreme Programming

Extreme Programming es una metodología de desarrollo ágil. Se diferencia de las metodologías tradicionales principalmente porque pone más énfasis en la adaptabilidad que en la previsibilidad, considerando que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. El principio de ***Extreme***

Programming es, que la metodología no es la ágil, sino más bien la metodología es la que ayuda a las personas a ser ágiles.

Quizás el mayor problema al desarrollar software son los cambios en las necesidades en el transcurso del proyecto. La mayoría de proyectos tienen requerimientos cambiantes, por lo que sería imposible garantizar que no se susciten cambios en el camino, sin dejar de lado que siempre va a existir ámbitos en los requisitos que no pueden cambiar.

Con ***Extreme Programming*** es posible producir software a partir de la primera semana de desarrollo, y de esta manera el cliente puede ir observando prototipos que le serán de ayuda para aclarar sus necesidades y de forma iterada se ingresa en un proceso de mejora y de afinación de lo que será el producto final. Esto genera un cambio de paradigma difícil de aplicar, pero a la vez abre nuevas vías para el desarrollo de software, dado que promueve cambios fundamentales en cómo manejar los proyectos.

Por ejemplo, en un proceso tradicional se planifican las actividades una tras otra, hasta la fecha de finalización del proyecto. El diseño siempre comienza cuando el análisis ha finalizado, y de igual manera empezará la codificación cuando el diseño esté listo, y así sucesivamente. Todo con el objetivo de hacer un plan para tener todo terminado a tiempo; a la finalización del proyecto se presenta el software desarrollado esperando que al cliente le guste.

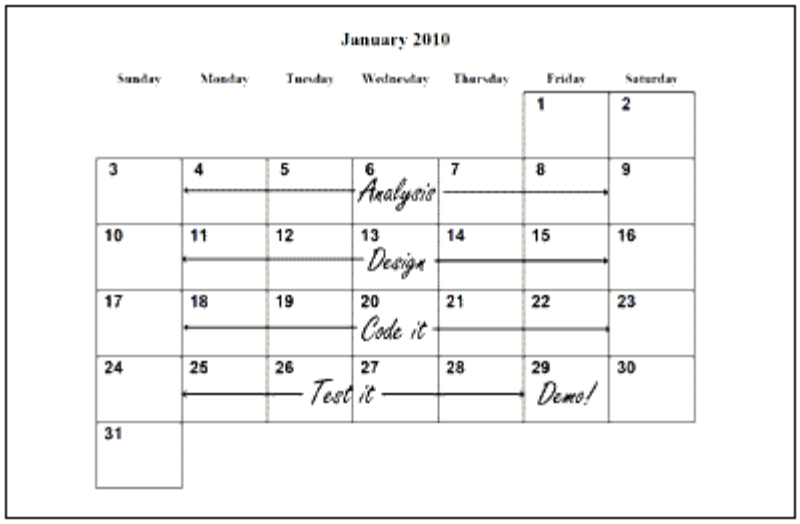

Figura 1 - 01: Forma tradicional de planificar [2]

En la Figura 1 -01 se puede visualizar que las necesidades permanecen constantes durante todo el proyecto, los requisitos forman un flujo a través de las diferentes actividades como una línea de producción. Este enfoque ofrece muchas desventajas en costos y grandes esfuerzos para lograr eficiencia en cada una de las etapas.

En cambio, la propuesta de ***Extreme Programming*** es que, al aceptar que los requisitos no son contantes se puede crear oportunidades de mejora y por ende productos de alta calidad. Cada función que quiere el cliente se la debe representar como una "historia del usuario", con lo cual los cambios tienen bajo costo dado que el alcance del proyecto será el certero y el presupuesto no fue afectado en el análisis completo por adelantado.

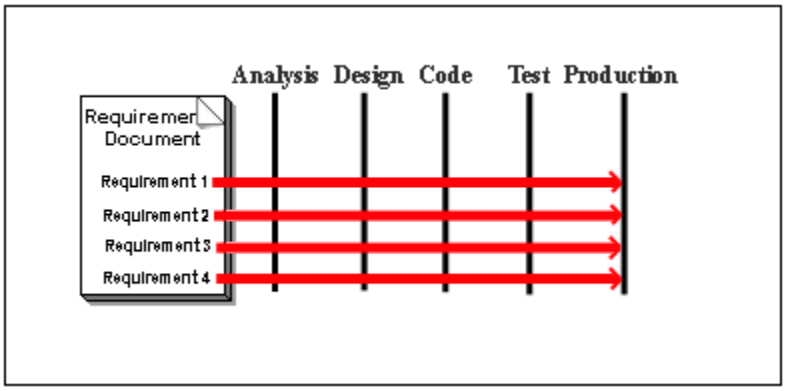

Figura 1 - 02: Fases propuesta por Extreme Programming [2]

Si se invierte el eje de la Figura 1 - 02 se visualizará a las actividades, el propio proceso, como la constante. El proceso se aplica a las "historias de usuario" en secuencia, las actividades están en curso y las "historias de usuario" son las metas cortas a cumplir.

Parece complicado pero con los ingenieros de software profesionales en cada proyecto es posible relajarse sabiendo que el equipo hará lo necesario para realizar el trabajo.

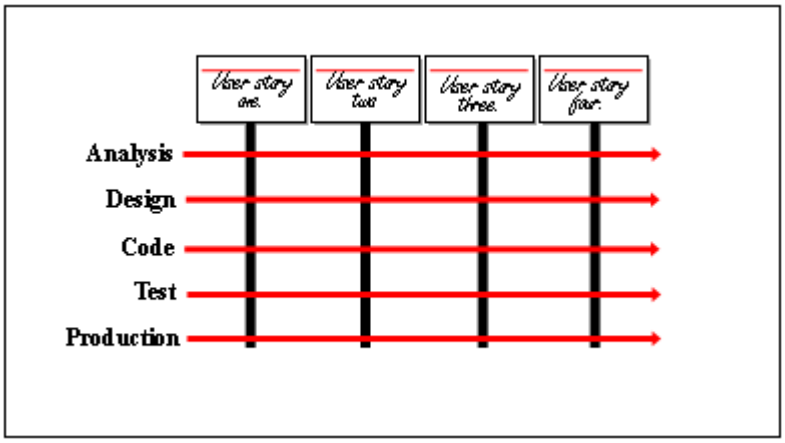

Figura 1- 03: Las verdaderas constantes del proyecto [2]

Al momento de utilizar ***Extreme Programming*** el calendario contendrá las actividades en orden de importancia como se muestra en la Figura 1- 04. Esto ofrece una visión diferente para ser eficientes, y poder cambiar el rumbo sin incurrir en grandes costos. Las iteraciones sobre los sistemas parciales ayudan a detectar grandes cambios antes de que sean caros.

Por lo tanto, un proceso ágil toma el proceso tradicional y lo convierte a noventa grados de lado. Esto permite a los administradores obtener un costo estimado por función en lugar de por actividad. Los clientes pueden tomar las decisiones difíciles sobre lo que puede ser dejado de hacer de una manera inteligente.

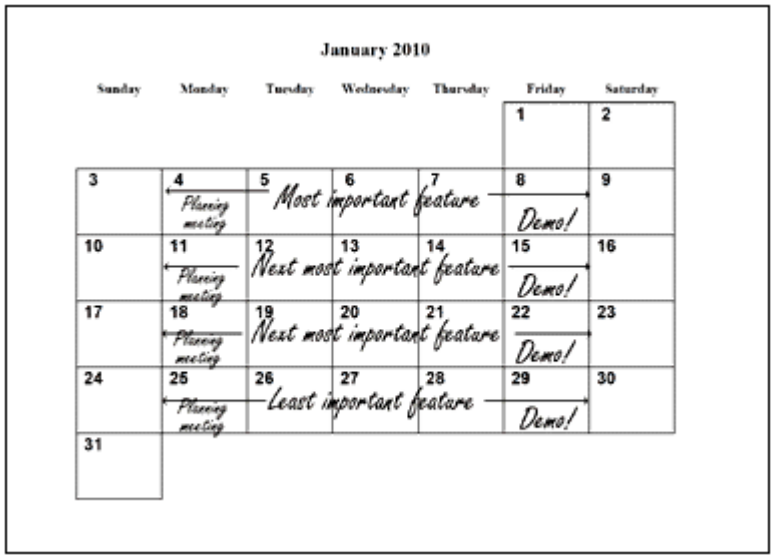

Figura 1 - 04: Planificación propuesta por Extreme Programming [2]

En este punto es importante identificar si el software diseñado para ser simple y elegante es más valioso que el software que es complejo y difícil de mantener. Para mantener la agilidad se debe acudir al arte de hacer cambios de diseño a través del tiempo, y de esta manera

mantener el software alineado a los procesos de las empresas y en la espera de los nuevos cambios. Un proceso ágil funciona igual de bien tanto en la fase de desarrollo como en el mantenimiento de software para facilitar la transición.

Nada de lo anteriormente mencionado es posible sin poner los pies sobre la tierra. Es importante ser honestos en las estimaciones, utilizando un proceso de planificación que debe ser acordado con el cliente para guiar nuestras decisiones en cuestión de costos. También la calidad del código fuente tiene suma importancia y por tanto debe mantener estándares y baja complejidad.

Es preferible decir planificación iterativa en lugar de desarrollo iterativo, ya que de esta manera se pone énfasis en el lugar que corresponde. Es posible planear todo el proyecto en iteraciones detalladas por adelantado, pero estaríamos atentando con el principio de Agilidad.

En la lista de actividades por desarrollar se puede ir eligiendo entre las más importantes. Importante significa diferentes cosas para diferentes equipos, pero el riesgo es una forma para empezar a discriminar.

La colección de las características más importantes es la entrada al proceso de planificación, las cuales son representadas de una manera mínima, que permite obtener estimaciones de esfuerzo, pero no detalles. Es innecesario perder tiempo en cada uno de los detalles dado que es probable que sólo se aplique una parte de las características en el proyecto, y por ello muchos equipos ágiles usan una forma rápida de generar alcances del proyecto llamadas "tarjetas de historia".

Las "tarjetas de historia" son la moneda de cada proyecto, al igual que el dinero que no tiene valor por sí mismo por ser un pedazo de papel, pero que representan un valor y puede ser utilizado para comprar cosas, las historias representan software valioso a los clientes, el tiempo a los desarrolladores, y los incrementos negociables a los administradores. Historias sin valor para el cliente sería como tener billetes falsos.

Existen tres niveles de planificación ágil.

- **Planificación de lanzamiento**.- es un grupo de historias seleccionadas porque representan un conjunto de características útiles que pueden ser solucionadas juntas. Este tipo de planes se hacen mediante la selección de las historias, y decidiendo cuántas iteraciones son necesarias o mediante la selección de una fecha de lanzamiento para ver cómo se puede ir iterando a partir de dicho lanzamiento. Los planes de lanzamiento no tienen información que no sea una lista de historias por hacer antes de una fecha.

- **Plan de iteración**.- este plan es un subconjunto de las historias que se llevarán a cabo en la iteración siguiente, y solo existe un plan de iteración a la vez. Con la colección escogida de las características importantes se puede estimar la cantidad de esfuerzo para ponerlas en práctica. La gente que va a hacer el trabajo, los desarrolladores, tienen autoridad para fijar las estimaciones. El administrador establecerá la cantidad total de trabajo que la siguiente iteración puede tener y el cliente elige un subconjunto de las características más importantes que se ajuste a la siguiente iteración. En este

nivel los casos de uso también podrían ser creados. Este mayor nivel de detalle es permitido porque las iteraciones son muy breves.

- **Plan diario**.- mediante una pequeña reunión diaria se procede a presentar el plan trabajo diario para luego actuar en consecuencia. Está permitido un mayor detalle porque la duración del plan es de un día y nada más.

Es importante mantener un ritmo sostenible durante todo el proyecto. El llegar antes de lo previsto ayuda a empezar con nuevas tareas, sin embargo en algún momento la demora es agradable, pero hay que mantener a su equipo comprometido para que no se retrasen a sí mismos.

Cuando un cliente se entera de algo importante acerca de los requisitos o encuentra una ventaja competitiva en la industria, los procesos cambian y al mismo tiempo el software debe adaptarse a este nuevo cambio para seguir cumpliendo su función. Los cambios en los requisitos son bienvenidos dado que previamente se ha decidido que se va a realizar nuevos planes de todos modos.

Los cambios que aparecen en un proyecto son lo más valioso ya que es el momento en donde se genera la posibilidad de acercarse más a la solución y/o automatización del dominio del problema. La planificación iterativa es hacer un plan, construir algún tipo de software, y luego hacer otro plan en base a lo que se ha aprendido.

La única opción para conocer la cantidad de tiempo que debe tomar cada iteración es haciendo planes honestos. Los procesos agiles tienden a apoyar las estimaciones honestas al aceptar las capacidades y limitaciones en su forma natural, por lo tanto, si se comete un error en un plan se lo acepta analizando la magnitud del mismo y esto servirá como experiencia para la creación de los nuevos. Es importante recalcar que no hay culpa, no hay sensación de fracaso, únicamente ha surgido la posibilidad de aprender algo nuevo.

Usualmente se realizan las estimamos para el desarrollo de software utilizando unidades que no son familiares como horas, días y semanas, pero realmente no se conoce cuánto tiempo es. Los administradores de proyectos se enteran de dichas medidas sólo después de la primera medición del progreso real, y por lo tanto no hay nada malo con el uso de unidades de tiempo, siempre y cuando todos los involucrados conozcan que éste no puede ser medido por el reloj de la pared.

La velocidad será un parámetro importante en el proceso de planificación iterativa. Calcular la velocidad mediante la suma de horas, días para determinar el tiempo que toma cada iteración, de tal modo que, el director del proyecto pueda usar esta información para sugerir la cantidad de trabajo que se podrá realizar en la siguiente iteración. Es importante tener en cuenta que la primera iteración no puede tener una velocidad.

Todas las personas aportan algo diferente en cada equipo, si suponemos que la media de eficiencia de un equipo que cuenta con diez integrantes será el 10% de la capacidad, es

simplemente equivocado. Comparar la velocidad entre dos equipos también es una mala idea puesto que cualquiera de los dos equipos tendrá velocidades diferentes.

Entonces, los "planes honestos" son algo más que las estimaciones de la verdad; estos están estrechamente relacionados con una fecha de caducidad o ***deadline***, por lo tanto es indispensable ser honesto acerca de cuánto tiempo puede permanecer un plan vigente y cuando ya es necesario crear uno nuevo.

Adicionalmente, todo proyecto debe tener un ***HeartBeat***[1] constante, con iteraciones de longitud fija y tan cortas como sea posible. Un ***HeartBeat*** de trabajo produce software listo para su despliegue en producción y creará un nuevo plan basado en la retroalimentación de la iteración anterior y en el cambio de las necesidades del cliente, nunca en lo que quedó sin terminar en la iteración anterior. Siempre para empezar con una nueva iteración la anterior debió haber sido concluida.

El ingrediente esencial de los cambios en el desarrollo iterativo es la retroalimentación. Una iteración más corta proporcionará información más frecuente y por consiguiente se obtendrá más oportunidades de aprender. La comunicación diaria también toma un papel importante, dado que todo el equipo debe sentirse involucrado en el proyecto. Una reunión diaria a la misma hora y en el mismo lugar es una forma de aportar al *Heartbeat* constante. Durante las reuniones deben existir al menos tres preguntas: ¿qué hiciste ayer?, ¿qué vas a hacer hoy?, y

[1] Ritmo constante durante el desarrollo de un proyecto. Este término es utilizado por su analogía con el latido del corazón.

¿qué está bloqueando su progreso?. Al mantener una breve reunión no existirá el problema con la participación de los integrantes del equipo.

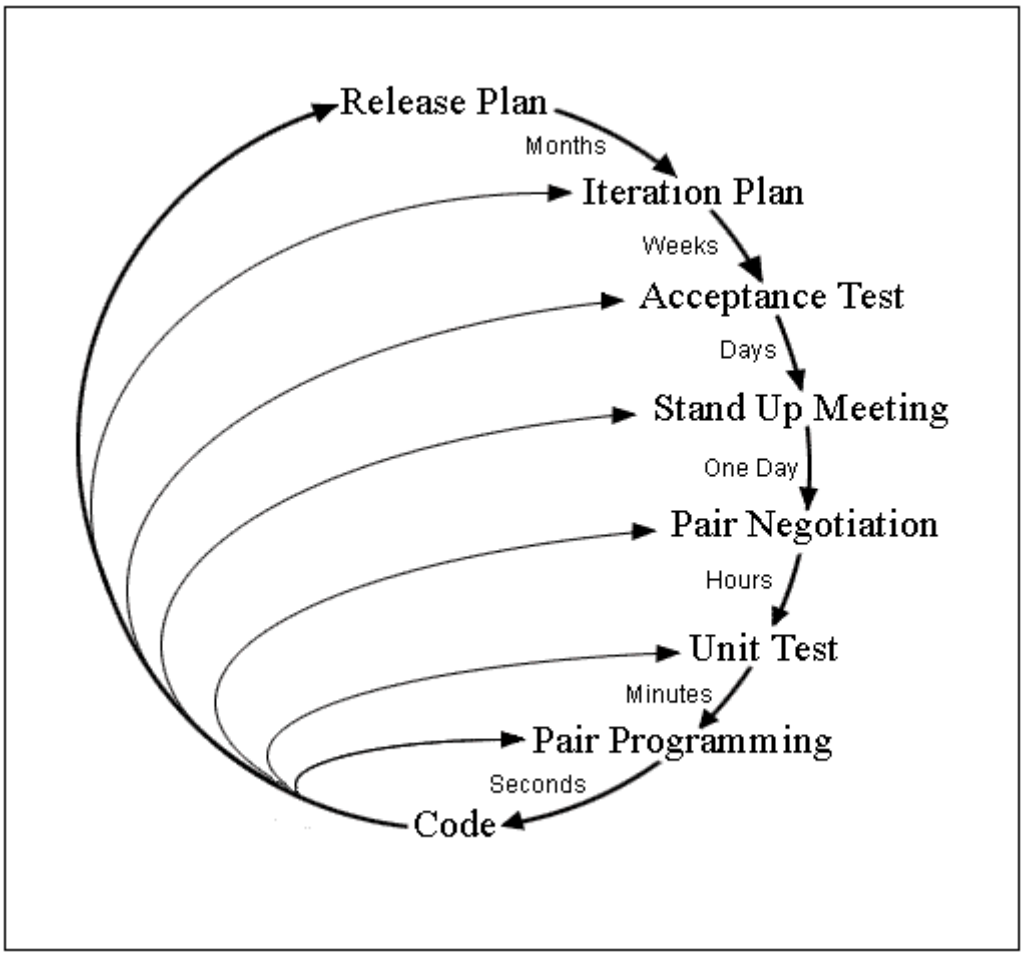

Figura 1 - 05: Planificación y Retroalimentación [2]

En la Figura 1 – 05 se puede apreciar los circuitos de retroalimentación o canales de comunicación recomendado por ***Extreme Programming***. Estos canales de comunicación están dispuestos en una espiral para mostrar la frecuencia de cada uno de los ítems.

No se debe desperdiciar tiempo exponiendo las ideas a demasiado detalladas frente a la audiencia equivocada, y en el caso de encontrarse explicando los detalles en el nivel superior hay que preguntarse el por qué. Si se siente la necesidad de explicar puede ser por estar tratando de justificar las acciones tomadas.

Dado que se cuenta con planes honestos es posible aceptar errores de forma honesta, aprender de los errores y hacer un nuevo plan para seguir adelante, retirando la culpa del proyecto para que todo el mundo hable libre y abiertamente. Las personas no necesitan una excusa para cometer un error sólo tiene que aprender de ellos.

"El cliente no sabía lo que quería" [C] es una excusa común para el fracaso del proyecto y que no es válida. El cliente sabe exactamente lo que quiere, una solución al problema en cuestión. Si el proyecto fracasa por el software mal construido, no es la culpa de los clientes, la culpa es de los desarrolladores. Los desarrolladores son los responsables de hacer las preguntas necesarias hasta lograr tener claro el panorama del problema y de esta manera ir en busca de una buena solución.

Los clientes son parte de los equipo del desarrollo de software, y no es correcto pensar que el tiempo de los expertos es demasiado valiosa para estar disponible como recursos en los proyecto.

1.2.1. Fases aplicadas en esta disertación

Las fases propuestas por Extremme Programming que se utilizarán en la presente disertación se exponen en la Tabla 1 – 01.

Fase	Disertación	Observaciones
Release Plan	✓	
Iteration Plan	✓	
Aceptance Plan		
Stand Up Meeting	✓	
Pair Negotiation	✓	
Unit Test	✓	
Pair Programming	✓	Programación individual

Tabla 1 - 01: Fases de Extremme Programming aplicadas en presente disertación.

1.3. Aplicaciones Web

La Web fue diseñada con el objetivo de facilitar los estudios científicos, dado que de esta manera era mucho más sencillo compartir información con el uso de documentos de hipertexto. Los documentos de hipertexto están creados usando HTML[2], el cual es un sistema de códigos que permiten colocar ***links*** a la información dentro de un mismo sitio. Dando origen a los sitios web, que son varias páginas HTML unidas por ***links***.

Las aplicaciones web en ingeniería de software son aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante

[2] Hypertext Markup Language

un ***Browser***. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web el cual es el encargado de su ejecución.

Este tipo de aplicaciones son populares debido a lo práctico del ***browser*** como cliente ligero, así como la facilidad para actualizar y mantener aplicaciones sin tener la necesidad de distribuir e instalar software en los miles de usuarios. Existen aplicaciones como los web mails, wikis, web blogs, tiendas en línea y la propia Wikipedia que son ejemplos bastante conocidos que nos dan una visión clara de las capacidades de los mismos.

Un ***Browser*** es un software que brinda la posibilidad a los usuarios para tener fácil acceso, visualización y navegación de páginas web. Entre los más utilizados se encuentran: Internet Explorer, Firefox, Safari, Opera, Lynx.

Los ***browsers*** presentan elementos de las páginas web de acuerdo a lo estipulado en el código HTML, dado que el código HTML provee al browser con información como: la fuente de la letra y su tamaño; la ubicación y el formato en los cuales se desplegarán los gráficos, animaciones, audio o videos; la ubicación de los ***hyperlinks***, y su destino; otro tipo de formatos como la alineación de contenidos, páginas y color de los textos, entre otros.

Es importante mencionar que una página web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en diversos juegos o acceder a gestores de base de datos de todo tipo.

El modo de generar páginas dinámicas ha evolucionado, desde la utilización del CGI[3]. Estas tecnologías se encuadran dentro de aquellas conocidas como ***Server Side***, ya que se ejecutan en el servidor web, y se complementan con la tecnología del lado del cliente, ***Client Side***, que se refieren a las posibilidades de que las páginas lleven incrustado código que se ejecuta en el cliente, como por ejemplo JavaScript.

El esquema general de las aplicaciones web se presenta en la Figura 1 - 06, en este se muestra cada tipo de tecnología involucrada en la generación e interacción de documentos Web.

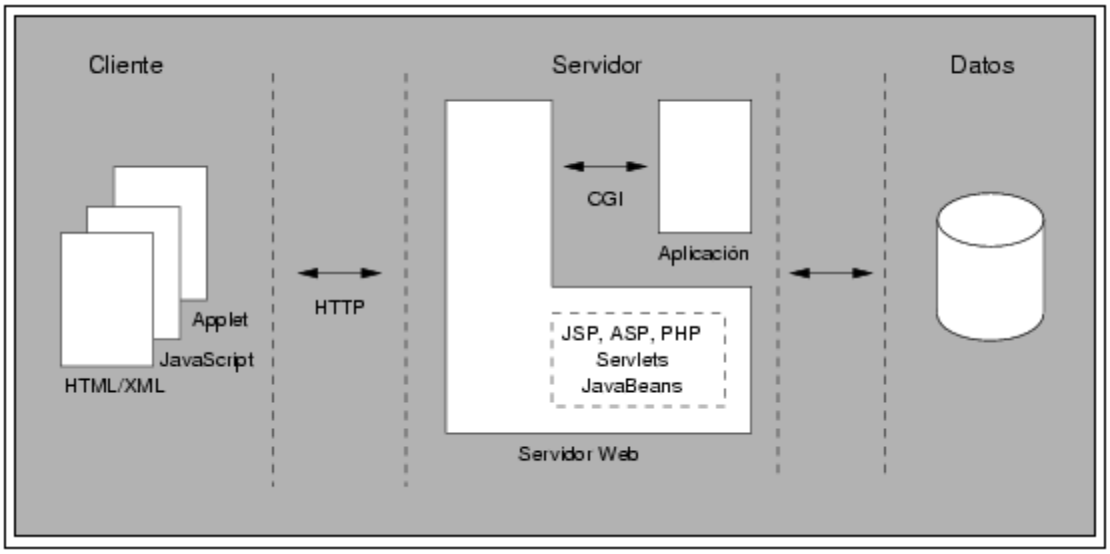

Figura 1 - 06: Esquema general de la tecnología web

A continuación se describe las principales características y funcionalidades de las diferentes tecnologías involucradas en el Web.

[3] Common Gateway Interface

1.3.1. World Wide Web

La WWW[4], fue creada en 1989 en *The European Particle Physics Laboratory* ubicado en Ginebra, Suiza.

La WWW es un sistema hipermedia interactivo desarrollado sobre Internet. La idea de hipermedia es la de juntar texto, imágenes, audio y vídeo dentro de un mismo envoltorio llamado documento. La cual se asienta sobre el protocolo HTTP[5] y sobre el lenguaje de definición de documentos hipertexto HTML.

Se basó en muchas fuentes para crear el WWW entre las más destacadas se encuentran: TCP/IP[6], Tipos MIME[7], SGML[8], Arquitecturas Cliente-Servidor, entre otros. De esta manera la WWW es el sistema de difusión del conocimiento más importante que implementa estas nociones.

[4] World Wide Web
[5] Hyper Text Transfer Protocol
[6] Protocolo de control de transmisión/Protocolo de Internet. Es un subconjunto de protocolos pero lleva el nombre de los dos más importantes.
[7] Multipurpose Internet Mail Extensions. Es una serie de convenciones o especificaciones dirigidas al intercambio a través de Internet de todo tipo de archivos (texto, audio, vídeo, etc.) de forma transparente para el usuario.
[8] Standard Generalized Markup Language

1.3.2. HyperText Transfer Protocol (HTTP)

A pesar de que entender el modo en que funciona HTTP no es estrictamente necesario para desarrollar aplicaciones Web, algunas nociones pueden ser útiles al momento de desarrollar este tipo de aplicaciones para hacerlo con más facilidad y confianza.

HTTP es un protocolo del nivel de aplicación para sistemas de información multimedia distribuidos, que puede ser utilizado para más propósitos que para manejar archivos HTML. Entre las propiedades de HTTP se pueden destacar las siguientes:

- *Un esquema de direccionamiento comprensible*. Utiliza el Universal Resource Identifier (URI) para localizar sitios (URL) o nombres (URN) sobre los que hay que aplicar un método. La forma general de un URL es *servicio://host/disertacion.ext.*

- *Arquitectura Cliente-Servidor*. HTTP se asienta en el paradigma solicitud/respuesta. La comunicación se asienta sobre TCP/IP y el puerto por defecto a utilizarse es el 80, pero es posible utilizar otros según las necesidades.

- *Es un protocolo sin conexión y sin estado*. Después de que el servidor ha contestado la petición del cliente, se rompe la conexión entre ambos. Además no se guarda en memoria el contexto de la conexión para siguientes conexiones.

- *Está abierto a nuevos tipos de datos*. HTTP utiliza tipos MIME para la determinación del tipo de los datos que transporta. Cuando un servidor HTTP transmite información de vuelta a un cliente, incluye una cabecera que le indica al cliente sobre los tipos de datos que componen el documento. De la gestión de esos datos se encargan las utilidades que tenga el cliente (visor de imágenes, de vídeo, etc.)

Una transacción HTTP está compuesta por una cabecera, y opcionalmente, por una línea en blanco seguida de los datos. En la cabecera se especifica tanto la acción solicitada en el servidor, como los tipos de datos devueltos o un código de estado.

1.3.3. La Interacción entre el *Browser* y el Servidor

Durante una sesión normal de trabajo en *WWW* un cliente (*browser*) solicita un documento de un servidor web y una vez obtenido lo muestra al usuario que hizo la solicitud. Si este documento contiene un enlace a otro documento en el mismo o en distinto servidor, y el usuario activa el enlace del cliente web efectuará otra petición y mostrará el nuevo documento.

Durante la comunicación entre el cliente y el servidor HTTP en el que el cliente solicita el documento *doc.html* al servidor se intercambian la siguiente transacción HTTP:

```
GET /doc1.html HTTP/1.0
Accept: www/source
Accept: text/html
Accept: image/gif
User-Agent:  Lynx/2.2  libwww/2.14
From: paguilar@puce.edu.ecs
/* esto es una linea en blanco */
```

Figura 1 - 07: Transacción HTTP [A]

El método GET indica el archivo que el cliente solicita y la versión de HTTP. El cliente también muestra una lista de los tipos MIME que puede aceptar como retorno, además de identificar el *browser* que utiliza (para que el servidor pueda optimizar los archivos para el tipo particular de navegador) y su dirección de correo electrónico. Al final existe una línea en blanco que determina el final de la cabecera HTTP. El servidor responde mandando la siguiente transacción HTTP:

```
HTTP/1.0  200 OK
Date: Friday, 23-Feb-01 16:30:00 GMT
Server: Apache/1.1.1
Content-type: text/html
Content-length: 230
/* esto es una linea en blanco */
<HTML><HEAD><TITLE> ..... </HTML>
```

Figura 1 - 08: Respuesta HTTP del servidor [A]

En este mensaje el servidor utiliza la versión 1.0 de HTTP, y manda el código de estado 200 para indicar que la petición del cliente ha sido procesada satisfactoriamente. También se

identifica como un servidor Apache. Indica al cliente que el contenido del documento es texto en formato HTML y que tiene una longitud de 230 bytes.

La primera línea de una petición contiene los comandos HTTP, conocidos como métodos. Los más conocidos y utilizados son tres: GET, HEAD y POST.

El método GET se utiliza para recuperar información identificada por un URI por parte de los navegadores. Si el URI se refiere a un proceso generador de datos como un programa CGI, en lugar de él, se devuelven los datos generados por el programa. El método GET también se puede utilizar para pasar una pequeña cantidad de información al servidor en forma de pares atributo-valor añadidos al final del URI detrás de un símbolo de interrogación, "?".

```
GET /cgi/saludar.pl?nombre=paolo&email=paguilar@puce.edu.ec HTTP/1.0
```

Figura 1 - 09: Método GET [A]

La longitud de la petición GET está limitada por el espacio libre en los *buffers* de entrada. Por lo que para mandar una gran cantidad de información al servidor se debe utilizar el método POST.

El método HEAD es idéntico al GET excepto que el servidor no devolverá el cuerpo del mensaje en la respuesta a un método HEAD. Esto es útil para obtener información sobre las

entidades implicadas en la petición sin que tengan que transferirse. Sirve para comprobar si los enlaces son válidos o para saber cuándo fue la última modificación de la entidad solicitada.

El método POST se refiere normalmente a la invocación de procesos que generan datos que serán devueltos como respuesta a la petición. Además se utiliza para aportar datos de entrada a esos programas. En este caso los pares atributo-valor son incluidos en el cuerpo de la petición separados por *ampersand*.

```
POST /cgi/saludar.pl HTTP/1.0
Accept: */*

nombre=paolo&email=paguilar@puce.edu.ec
```

Figura 1 - 10: Método POST [A]

De este modo el método POST no tiene limitaciones de espacio y puede enviar mucha más información al servidor.

2. CAPITULO II: Servicios de soporte y mantenimiento

En este capítulo se tratarán los temas involucrados en la funcionalidad del sistema, los mismos que serán de ayuda al momento de definir la funcionalidad de la aplicación web. Es importante tener conocimiento sobre los servicios de soporte y mantenimiento para acercarme a las necesidades globales con las que cuentan las empresas que desenvuelven estas actividades.

2.1. Gestión de servicios

La definición de Servicios según diversos expertos:

- Stanton, Etzel y Walker, definen los servicios "como actividades identificables e intangibles que son el objeto principal de una transacción ideada para brindar a los clientes satisfacción de deseos o necesidades" [D]. En esta propuesta, cabe señalar que según los mencionados autores ésta definición excluye a los servicios complementarios que apoyan la venta de bienes u otros servicios, pero sin que esto signifique subestimar su importancia.

- Para Richard L. Sandhusen, "los servicios son actividades, beneficios o satisfacciones que se ofrecen en renta o a la venta, y que son esencialmente intangibles y no dan como resultado la propiedad de algo" [E].

- Según Lamb, Hair y McDaniel, "un servicio es el resultado de la aplicación de esfuerzos humanos o mecánicos a personas u objetos. Los servicios se refieren a un hecho, un desempeño o un esfuerzo que no es posible poseer físicamente" [F].

- Para la American Marketing Association (A.M.A.), los servicios son "productos, tales como un préstamo de banco o la seguridad de un domicilio, que son intangibles o por lo menos substancialmente. Si son totalmente intangibles, se intercambian directamente del productor al usuario, no pueden ser transportados o almacenados, y son casi inmediatamente perecederos. Los productos de servicio son a menudo difíciles de identificar, porque vienen en existencia en el mismo tiempo que se compran y que se consumen. Abarcan los elementos intangibles que son inseparabilidad; que implican generalmente la participación del cliente en una cierta manera importante; no pueden ser vendidos en el sentido de la transferencia de la propiedad; y no tienen ningún título. Hoy, sin embargo, la mayoría de los productos son en parte tangibles y en parte intangibles, y la forma dominante se utiliza para clasificarlos como mercancías o servicios (todos son productos). Estas formas comunes, híbridas, pueden o no tener las cualidades dadas para los servicios totalmente intangibles" [3].

- Kotler, Bloom y Hayes, definen un servicio de la siguiente manera: "Un servicio es una obra, una realización o un acto que es esencialmente intangible y no resulta necesariamente en la propiedad de algo. Su creación puede o no estar relacionada con un producto físico [F]. Complementando esta definición, cabe señalar que según los mencionados autores, los servicios abarcan una amplia gama, que va desde el alquiler de una habitación de hotel, el depósito de dinero en un banco, el viaje en avión a la visita a un psiquiatra, hasta cortarse el cabello, ver una película u obtener asesoramiento de un abogado. Muchos servicios son intangibles, en el sentido de que no incluyen casi ningún elemento físico, como la tarea del consultor de gestión, pero otros pueden tener un componente físico, como las comidas rápidas [G].

Teniendo en cuenta las anteriores propuestas, se plantea a modo de resumen la siguiente definición de servicios: "Los servicios son actividades identificables, intangibles y perecederas que son el resultado de esfuerzos humanos o mecánicos que producen un hecho, un desempeño o un esfuerzo que implican generalmente la participación del cliente y que no es posible poseer físicamente, ni transportarlos o almacenarlos, pero que pueden ser ofrecidos en renta o a la venta; por tanto, pueden ser el objeto principal de una transacción ideada para satisfacer las necesidades o deseos de los clientes" [A].

La gestión de servicios actualmente tiene gran importancia en nuestra sociedad debido a que las empresas grandes han logrado producir bienes al por mayor con mayor eficiencia, y lo hacen a costos muy inferiores. Por lo tanto esta es la época del conocimiento, y de aquellos que brindan servicios.

El sector de servicios mantiene una tendencia incremental y cada día son más importantes en la economía mundial. Muchas naciones en desarrollo logran fortalecer su comercio gracias a que el *sector servicios* ha tomado auge. Todo esto ha permitido que las economías se transformen de una visión nación-nación a profesional-nación.

En el ambiente de los servicios se aplican directamente las técnicas utilizadas en la manufactura ya que prácticamente existe una analogía entre ambos, pero se debe tomar en cuenta que existen algunas diferencias específicas, las cuales permiten una gerencia innovadora y creativa de los mismos. Es decir, las mismas herramientas gerenciales utilizadas en la industria manufacturera son aplicables a los servicios.

En los servicios se debe realizar una distinción entre las entradas (clientes) y los recursos. La presencia del cliente como un participante en el proceso del servicio requiere atención al diseño lo cual no es encontrado en las operaciones tradicionales de fabricación. El cliente puede tomar una parte activa en el proceso y esta es una consideración importante.

Los servicios de acuerdo a *Roger Schmenner* pueden clasificarse en dos dimensiones que afectan significativamente el carácter de la entrega del mismo: el grado de la intensidad de labor, y el grado de interacción y personalización. El gerente de servicios en cualquier dimensión enfrentará retos similares.

La identificación del servicio es difícil de realizar debido a la naturaleza intangible de los mismos. Se puede decir que el paquete de servicio está compuesto por bienes y servicios que se proveen en algún ambiente. Este paquete consiste de las siguientes características:

- *Factibilidad de soporte,* es el recurso físico que debe estar en el sitio antes de ofertar el servicio.
- *Bienes para facilitar,* es el material consumido o adquirido por el comprador o los ítems que provee un cliente.
- *Información,* son los datos de operación o información que provee el cliente para habilitar un servicio eficiente y personalizado.
- *Servicios explícitos,* son los beneficios observables por los sentidos y están conformados por las características esenciales de un servicio.
- *Servicios implícitos,* son los beneficios psicológicos que el consumidor puede sentir solo vagamente o las características esenciales del servicio.

Todas estas características son experimentadas por el cliente y forman la base de la percepción del servicio.

También es importante tener en cuenta la eficiencia y la efectividad de la entrega de los servicios. Los servicios son creados y consumidos simultáneamente por lo que no son almacenados, y esta es una característica crítica en el proceso de gestión de servicios. Esto disminuye las posibilidades de control de calidad.

Los contratos de servicios tienen la oportunidad de construir relaciones a largo plazo con los clientes, ya que estos son transmitidos directamente, y a menudo en persona. Mientras los fabricantes tradicionalmente son aislados del usuario final por los canales de distribución que

están conformados por distribuidores y vendedores. Una ventaja competitiva es tener la oportunidad de conocer a los clientes.

2.2. Servicios de soporte informático

El soporte informático es el servicio mediante el cual los especialistas en apoyo informático proporcionan asistencia técnica, soporte remoto y asesoramiento a individuos y organizaciones que dependen de la tecnología de la información (TI). Existen dos tipos de soporte informático:

2.2.1. El servicio técnico informático que brindan su ayuda físicamente.

Responden a los usuarios de las organizaciones y ejecutan automáticamente los programas de diagnóstico para resolver problemas. Además, pueden escribir manuales y capacitar en el uso de hardware y software. En este tipo de soporte también se supervisa el funcionamiento diario de los sistemas informáticos de las empresas, la resolución de problemas técnicos con redes de área local (LAN), redes de área amplia (WAN), entre otros.

2.2.2. Soporte informático remoto o asistencia remota.

En este caso el servicio es prestado mediante llamadas telefónicas y/o mensajes de correo electrónico de los clientes que requieren el servicio. Para prestar este tipo de servicio los especialistas utilizan software que permite tener acceso al sistema en cuestión para poder diagnosticar la naturaleza del problema. De igual manera instalar, modificar, limpiar y reparar el software. Esta tarea es tan simple y económica gracias a las aplicaciones que permiten al técnico obtener control remoto del equipo y observar la pantalla del PC remoto con una conexión a escritorio remoto por Internet. Existen aplicaciones gratis que ofrecen este servicio como Logmein[9] o TeamViewer[10].

Este tipo de tarea es muy atractiva para las empresas dado que permite la reducción de costos de infraestructura y demás. El usuario solicita el soporte técnico y en poco tiempo recibe la ayuda solicitada. Este tiempo varía según el SLA[11] de los contratos.

Las ventajas que ofrece este servicio son: Ofrecer varias modalidades para acomodarse a las necesidades del cliente; Proveer especialistas entrenados para ofrecer una respuesta rápida y segura a los problemas; Generalmente es un servicio integrado a otros servicios contratados; y, Proveer un servicio intuitivo y seguro.

[9] Conjunto de servicios de software que proporciona acceso remoto a los ordenadores a través de Internet.
[10] Paquete de programas informáticos para el control remoto, compartir escritorio, y la transferencia de archivos entre ordenadores
[11] A**cuerdo de nivel de servicio** o ***Service Level Agreement***, es un contrato escrito entre un proveedor de servicio y su cliente con objeto de fijar el nivel acordado para la calidad de dicho servicio.

2.3. Servicios de mantenimiento informático

El mantenimiento informático es una de las actividades más comunes en el portafolio de servicios dado que es el proceso de mejora y optimización de equipos tecnológicos después de la entrega al usuario final.

Existen cuatro tipos reconocidos de operaciones de mantenimiento, los cuales están en función del momento del tiempo en el que se realizan, el objetivo particular para el cual son puestos en marcha, y en función de los recursos utilizados.

2.3.1. Mantenimiento Correctivo

Este tipo de mantenimiento es aplicado luego de que ocurre una falla o avería, es decir, solo tiene sentido cuando se presenta un error en el sistema. En este caso si no se produce ninguna falla, el mantenimiento será nulo, por lo que se tendrá que esperar hasta que se presente el desperfecto para recién tomar medidas de corrección de errores.

Este mantenimiento trae consigo las siguientes consecuencias: paradas no previstas en el proceso productivo, disminuyendo las horas operativas; afecta las cadenas productivas, es decir, que los ciclos productivos posteriores se verán parados a la espera de la corrección de la etapa anterior; presenta costos por reparación y repuestos no presupuestados, por lo que se dará el caso que por falta de recursos económicos no se podrán comprar los repuestos en el

momento deseado; la planificación del tiempo que estará el sistema fuera de operación no es predecible.

2.3.2. Mantenimiento Preventivo

Este tipo de mantenimiento también es denominado "mantenimiento planificado". Es aplicado antes de que ocurra una falla o avería, se efectúa bajo condiciones controladas sin la existencia de algún error en el sistema.

El mantenimiento preventivo es producido según la experiencia y habilidad del personal a cargo, los cuales son los encargados de determinar el momento necesario para llevar a cabo dicho procedimiento; el fabricante también puede estipular el momento adecuado a través de los manuales técnicos.

Se caracteriza por ser realizado en momentos en los cuales no se esta produciendo, por lo que se aprovecha las horas ociosas de la planta; se lleva a cabo siguiendo un programa previamente elaborado donde se detalla el procedimiento a seguir, y las actividades a realizar, a fin de tener las herramientas y repuestos necesarios "a la mano"; cuenta con una fecha programada, además de un tiempo de inicio y de finalización preestablecido y aprobado por los directiva de la empresa; está destinado a un área en particular y a ciertos equipos específicamente; permite a la empresa contar con un historial de todos los equipos, además

brinda la posibilidad de actualizar la información técnica de los equipos; y, permite contar con un presupuesto aprobado por la directiva.

2.3.3. Mantenimiento Predictivo

Este tipo de mantenimiento consiste en determinar en todo instante la condición técnica (mecánica y eléctrica) real de la máquina examinada, mientras se encuentre operativa, para ello se hace uso de programas que permiten métricas de los parámetros más importantes de los equipos.

El sustento tecnológico de este mantenimiento consiste en la aplicaciones de algoritmos matemáticos agregados a las operaciones de diagnóstico, los cuales brindan información referente a las condiciones del equipo.

Tiene como objetivo disminuir las paradas por mantenimientos preventivos, y de esta manera minimizar los costos por mantenimiento y por no producción. Sin embargo la implementación de este tipo de métodos requiere de inversión en equipos, en instrumentos, y en contratación de personal calificado.

Las técnicas más conocidas para la estimación del mantenimiento predictivo son: Analizadores de Fourier, para realizar análisis de vibraciones; Endoscopia, para observar lugares ocultos; Termovisión, para la detección de condiciones a través de la temperatura

desplegado; Medición de parámetros de operación como viscosidad, voltaje, corriente, potencia, presión, temperatura, entre otros.

2.3.4. Mantenimiento Proactivo

Este tipo de mantenimiento tiene como fundamento los principios de solidaridad, colaboración, iniciativa propia, sensibilización, trabajo en equipo, de tal modo que todos los involucrados directa o indirectamente en la gestión del mantenimiento deben conocer la problemática del mantenimiento, es decir, que tanto técnicos, profesionales, ejecutivos, y directivos deben estar consientes de las actividades que se llevan a cabo para desarrollas las labores de mantenimiento. Cada individuo desde su cargo o función dentro de la organización, actuará de acuerdo a este cargo, asumiendo un rol en las operaciones de mantenimiento bajo la premisa de que se debe atender las prioridades del mantenimiento en forma oportuna y eficiente.

El mantenimiento proactivo implica contar con una planificación de operaciones, la cual debe estar incluida en el *Plan Estratégico* de la organización. Este mantenimiento a su vez debe brindar indicadores (informes) hacia la gerencia, respecto del progreso de las actividades, los logros, aciertos, y también errores.

3. CAPITULO III: Análisis y Diseño

En este capítulo realizaré el diseño de lo que será la aplicación web, el mismo que no contendrá detalle a profundidad para no contradecir la metodología ágil propuesta en el capítulo uno.

Considero que es importante especificar los casos de uso, el diagrama de base de datos y el diseño de las pantallas dado que éstos servirán de pauta al momento de establecer las iteraciones para desarrollar el software.

Para realizar las pantallas de forma intuitiva para los usuarios los métodos ***CRUD***[12] estarán contenidos en la misma pantalla, es decir, sin tener que navegar a otras páginas será posible realizar las operaciones ***SQL*** [13] de forma ágil.

3.1. Definición del problema

La preocupación constante de las empresas es la gestión inteligente del flujo de la información de los contratos establecidos para los servicios de soporte y mantenimiento. Las empresas dedicadas a brindar servicios este tipo de servicios carecen de una gestión correcta

[12] Acrónimo de Crear, Obtener, Actualizar y Borrar (Create, Retrieve, Update y Delete en inglés)

[13] El lenguaje de consulta estructurado (por sus siglas en inglés *structured query language*) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas

de éstos, no cuentan con reportes, no existe una buena comunicación con los clientes, no llevan un buen control de la información relacionada, y muchas veces desconocen el estado en el que se encuentra el mismo.

Las empresas generalmente no cuentan con un proceso automatizado que les permita gestionar los servicios ofertados, lo cual genera dificultades al momento de administrarlos. No llevan un correcto historial de los servicios, ni control de su estado, siendo imposible obtener reportes a tiempo. Y como consecuencia, no pueden optimizar sus procesos para ser más ágiles.

Por lo tanto, se busca automatizar todo este proceso en un sistema web, mediante el cual, la empresa proveedora del servicio y sus clientes tendrán la posibilidad de acceder a la información requerida de forma íntegra y en tiempo real.

Adicionalmente, la empresa que desee implantar este software, lo podrá utilizar para mejorar la calidad de los servicios ofrecidos, logrando conocer el estado de la gestión de cada uno e incluyendo los nuevos para que sean atendidos de forma eficiente.

3.2. Casos de uso

Un caso de uso es una técnica aplicada para capturar los requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para

conseguir un objetivo específico. Normalmente, en los casos de usos se evita el empleo de jergas técnicas, y es preferible en su lugar un lenguaje más cercano al usuario final. En ocasiones, se utiliza a usuarios sin experiencia junto a los analistas para el desarrollo de casos de uso.

CODIGO	**CASO DE USO**
CU0	Ingresar al sistema
CU1	Gestión de usuarios
CU2	Gestión de empresas
CU3	Gestión de contratos
CU4	Gestión de roles de usuario
CU5	Gestión de tareas
CU6	Cambiar contraseña
CU7	Salir del sistema

Tabla 3 - 01: Casos de uso [A]

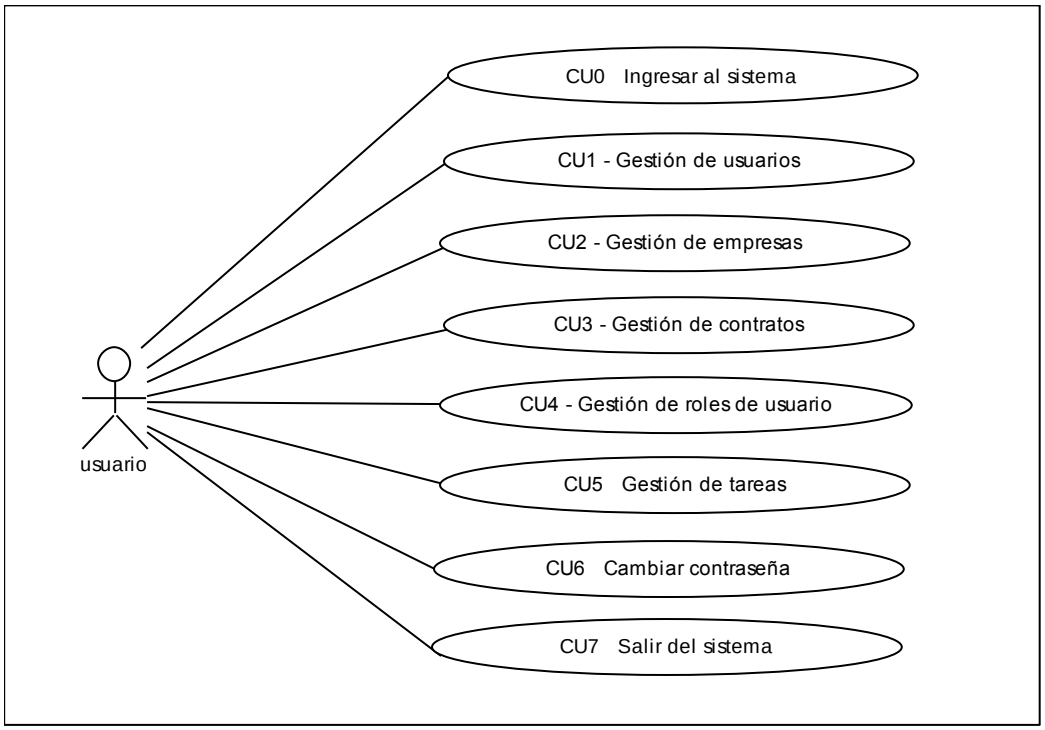

Figura 3 – 01: Casos de uso [A]

3.3. Modelo Entidad-Relación

El modelo Entidad-Relación es el modelo conceptual utilizado para el diseño de bases de datos. Fue introducido por Peter Chen en 1976. Está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas.

Este diseño se caracteriza por expresar entidades relevantes para un sistema de información así como sus interrelaciones y propiedades.

A continuación se expone el diagrama entidad relación correspondiente a esta disertación, a partir el cual se genera el script de la base de datos que se encuentra en el Anexo 1.

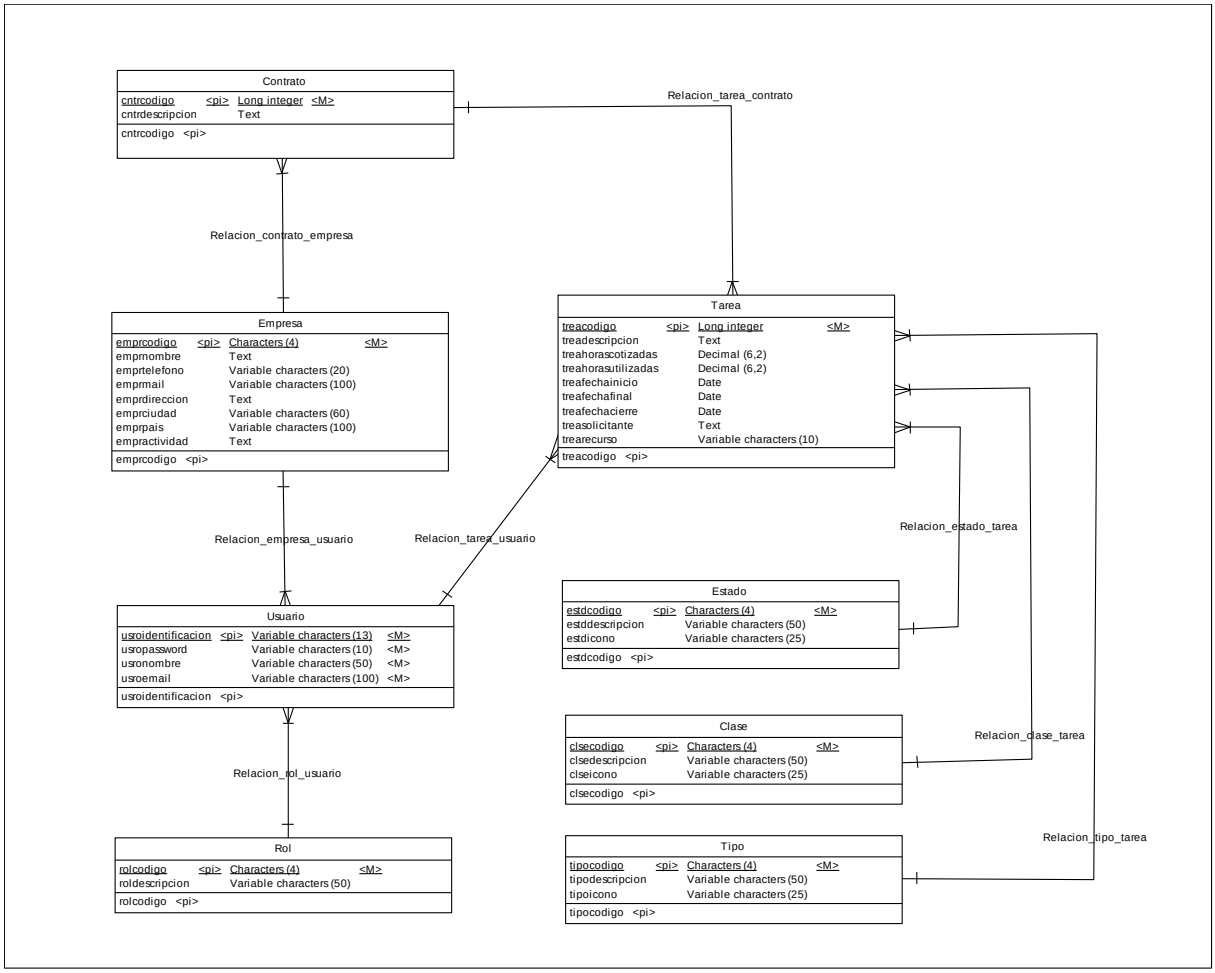

Figura 3 – 02: Diagrama conceptual Base de datos [A]

3.3.1. Diccionario de datos

Tabla Rol

CAMPO	DESCRIPCION	FUNCIONALIDAD
Rolcodigo	Código	Identificador único para los roles de usuario
Roldescripcion	Descripción	Descripción del rol para los usuarios

Tabla 3 – 02: Tabla Rol [A]

Tabla Usuario

CAMPO	DESCRIPCION	FUNCIONALIDAD
usroidentificacion	Cedula de identidad	Identificador de usuario
usropassword	Contraseña	Contraseña para acceder al sistema
usronombre	Nombre	Nombre completo del usuario
usroemail	Correo electrónico	Correo electrónico

Tabla 3 – 03: Tabla Usuario [A]

Tabla Empresa

CAMPO	DESCRIPCION	FUNCIONALIDAD
emprcodigo	Código	Identificador para las empresas

emprnombre	Nombre de la empresa	Razón social de la empresa
emprtelefono	Teléfono	Teléfono de la empresa
emprmail	Correo electrónico	Correo electrónico
emprdireccion	Dirección	Ubicación de las instalaciones
emprciudad	Ciudad	Ciudad de las instalaciones
Emprpais	País	País en el que se encuentra ubicada
empractividad	Actividad	Actividad que desempeña la empresa

Tabla 3 – 04: Tabla Empresa [A]

Tabla Contrato

CAMPO	DESCRIPCION	FUNCIONALIDAD
Cntrcodigo	Código	Identificador único
cntrdescripcion	Descripción	Descripción corta del contrato

Tabla 3 – 05: Tabla Contrato [A]

Tabla Tarea

CAMPO	DESCRIPCION	FUNCIONALIDAD
treacodigo	Código	Identificador único
treadescripcion	Descripción	Descripción de la tarea a realizar
treahorascotizadas	Horas cotizadas	Horas presupuestadas para la tarea solicitada

treahorasutilizadas	Horas utilizadas	Horas reales utilizadas
treafechainicio	Fecha inicial	Fecha de inicio
treafechafinal	Fecha de finalización	Fecha planificada para la finalización de la tarea.
Treacierre	Fecha real de cierre	Fecha real de finalización de la tarea
treasolicitante	Solicitante	Persona que solicita la tarea

Tabla 3 – 06: Tabla Tarea [A]

Tabla Estado

CAMPO	DESCRIPCION	FUNCIONALIDAD
estdcodigo	Código	Identificador único
estddescripcion	Descripción	Descripción del estado
estdicono	Icono	Imagen que representa el estado

Tabla 3 – 07: Tabla Estado [A]

Tabla Tipo

CAMPO	DESCRIPCION	FUNCIONALIDAD
tipocodigo	Código	Identificador único
tipodescripcion	Descripción	Descripción del tipo de tarea
tipoicono	Icono	Imagen que representa el estado

Tabla 3 – 08: Tabla Tipo [A]

Tabla Clase

CAMPO	DESCRIPCION	FUNCIONALIDAD
Clsecodigo	Código	Identificador único
clsedescripcion	Descripción	Descripción de la clase de servicio
Clseicono	Icono	Imagen que representa el estado

Tabla 3 – 09: Tabla Clase [A]

3.4. Diseño de pantallas

Pantalla 1. Ingreso al sistema (*Figura 3-03)*

Es la pantalla inicial del sistema, en la cual el usuario tendrá acceso al sistema mediante el número de identificación y contraseña.

Figura 3-03: Pantalla de ingreso al sistema [A]

Pantalla2. Inicio para el administrador del sistema (*Figura 3-04)*

En el caso de que el usuario tenga el rol de "Administrador" del sistema tendrá acceso al menú de gestión, en el cual se encuentran las opciones de Gestión de empresas, Gestión roles, Gestión de Usuarios, y acceso a los reportes.

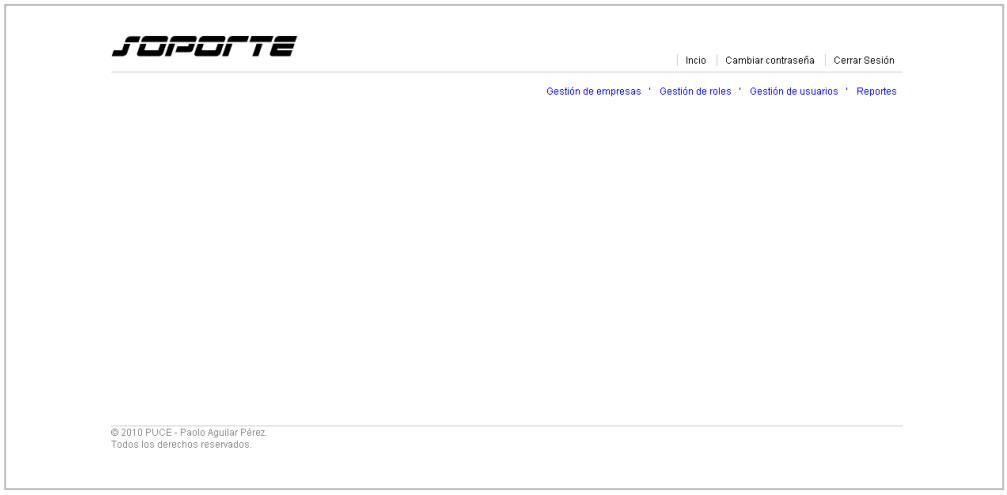

Figura 3-04: Pantalla inicial del administrador [A]

Pantalla 3. Inicio para los usuarios de consulta (*Figura 3-05)*

Esta es la pantalla inicial con la que interactuarán los usuarios de consulta con las cuales tendrán acceso únicamente a la información que les corresponde, dado sus tareas asignadas o su empresa.

Figura 3-05: Pantalla inicial de usuarios de consulta [A]

Pantalla 4. Pantalla tipo Gestión (*Figura 3-06)*

En estas pantallas será posible realizar las operaciones básicas de Inserción, Actualización y Eliminación sin la necesidad de navegar a abrir nuevas ventanas.

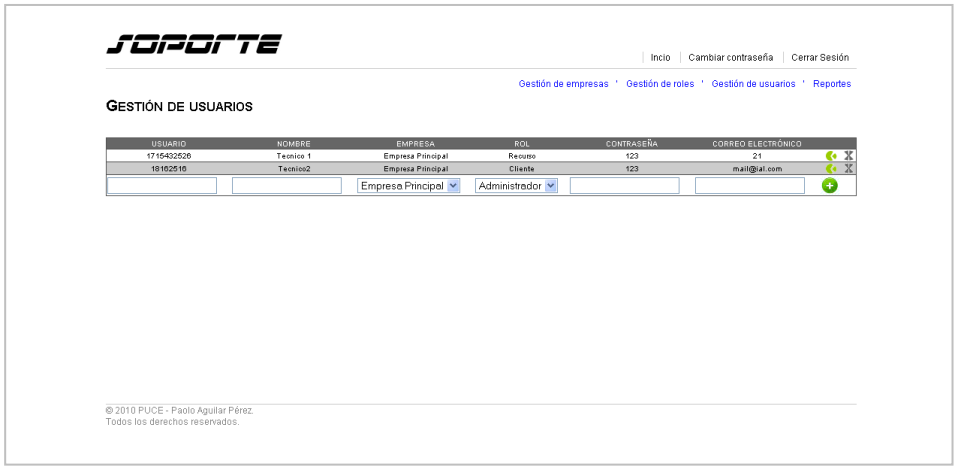

Figura 3-06: Pantalla tipo Gestión [A]

Pantalla 5. Pantalla de Reportes (*Figura 3-07)*

En esta pantalla se presentaran los reportes, los mimso que podrán ser exportados a un documento de Microsoft Excel para que puedan ser tabulados y de ser necesario se realicen gráficos estadísticos.

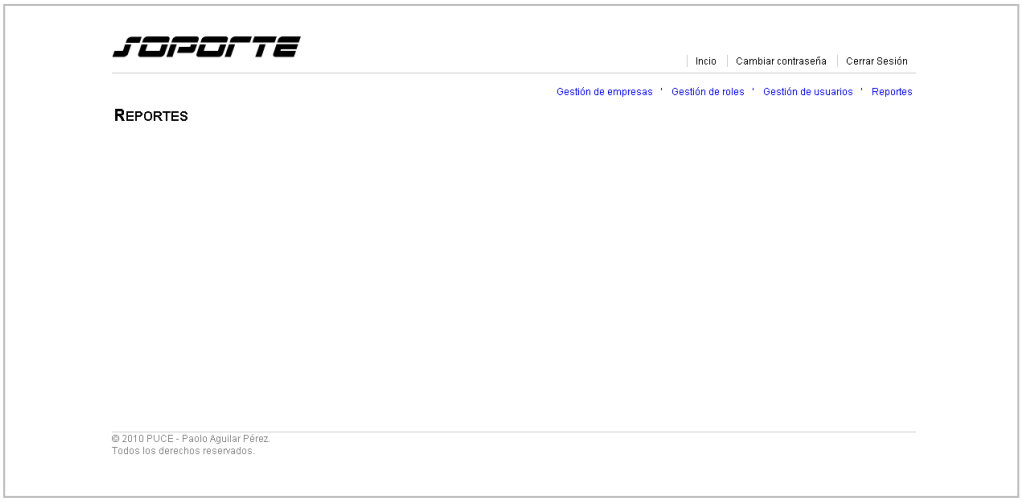

Figura 3-07: Pantalla Reportes [A]

4. CAPITULO IV: Construcción y retroalimentación

En este capítulo se explicará el proceso de desarrollo del sistema siguiendo la metodología utilizada de Extreme Programming. El software será desarrollado por iteraciones las cuales se encuentran clasificadas según funcionalidad, y culminarán una vez que se haya obtenido el producto requerido.

Es importante tener en cuenta que el proyecto debe tener un ***BeatHeart*** [14] constante y para empezar con una nueva iteración la anterior debe haber finalizado.

4.1. Diagrama de proceso

Antes de iniciar con las iteraciones propuestas por Extremme Programing se consideró importante describir el proceso que se va a automatizar para presentar de forma clara el procedimiento a seguir. A continuación en la Figura 4 -01 se describe el proceso global del sistema, el mismo que seguirá su curso dependiendo de los usuarios involucrados en cada una de las actividades.

[14] Ritmo constante durante el desarrollo de un proyecto. Este término es utilizado por su analogía con el latido del corazón.

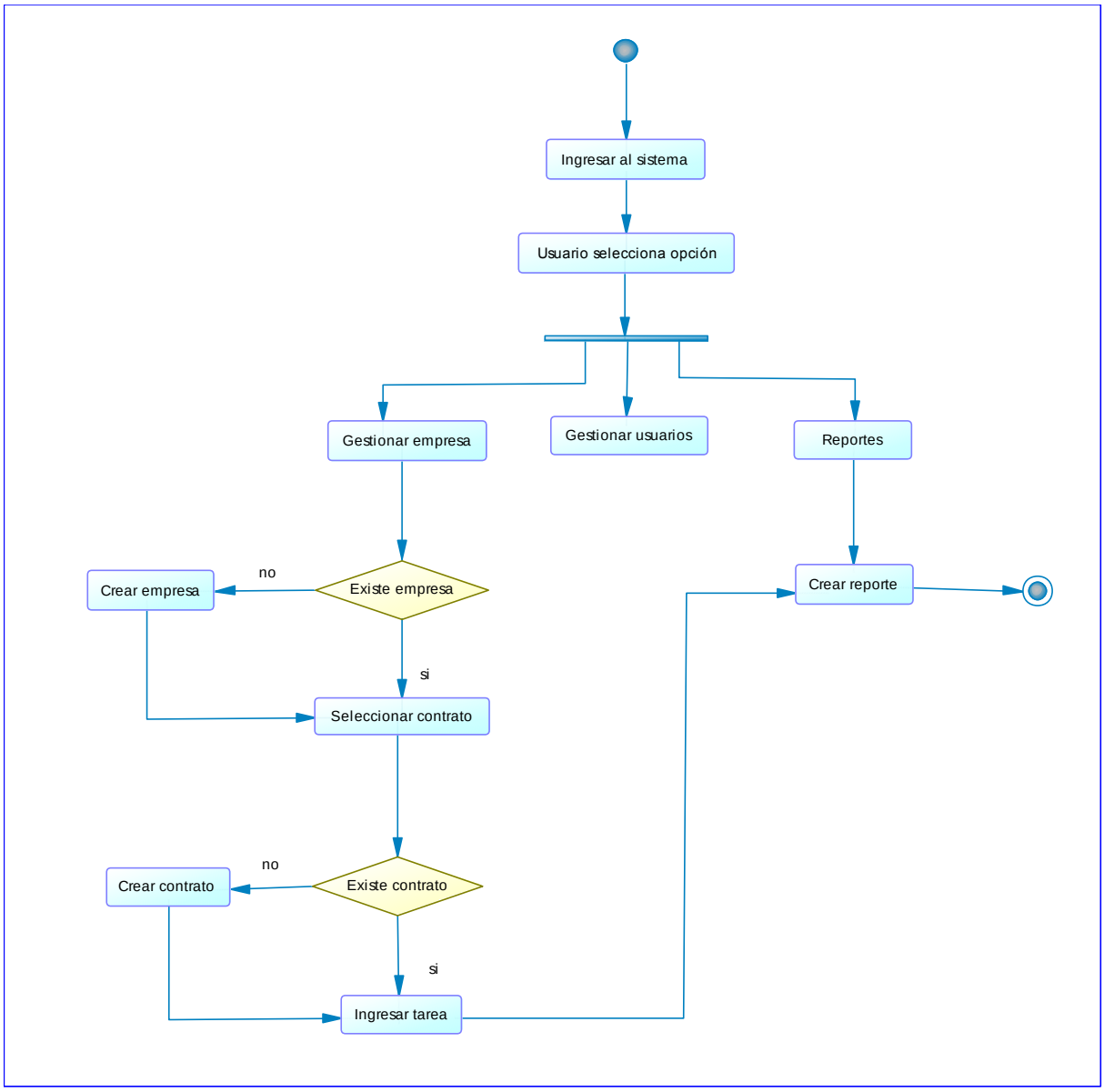

Figura 4-01: Diagrama de Proceso [A]

4.2. Iteraciones

Luego de lo revisado en los capítulos anteriores sobre Extreme Programming, los servicios de soporte y mantenimiento informático que dan origen a las respectivas tareas, y tener claro el proceso a automatizar se presenta las iteraciones a través de las cuales se irá construyendo el sistema.

4.2.1. Iteración No.1

Construcción de la gestión de datos del sistema y creación de la pantalla para el ingreso de usuarios.

- **Procedimiento**

Creación del script de la base de datos, utilizando la herramienta ***Power Designer,*** a partir del modelo conceptual realizado en la fase de diseño. Este script fue ejecutado en una Base de datos Microsoft SQL (Express Edition).

Una vez generada la base de datos generé el script de inicialización de la misma, el cual es indispensable para empezar a interactuar con el sistema, y se detalla a continuación. El script completo se encuentra en el Anexo 2.

```
--Creación de roles del sistema

INSERT INTO dbo.ROL VALUES ('0000' , 'Todos'         )
INSERT INTO dbo.ROL VALUES ('R001' , 'Administrador')
INSERT INTO dbo.ROL VALUES ('R002' , 'Cliente'       )
INSERT INTO dbo.ROL VALUES ('R003' , 'Técnico'       )
INSERT INTO dbo.ROL VALUES ('R004' , 'Adm. Empresas')
INSERT INTO dbo.ROL VALUES ('R005' , 'Adm. Roles'    )
INSERT INTO dbo.ROL VALUES ('R006' , 'Adm. Usuarios')
INSERT INTO dbo.ROL VALUES ('R007' , 'Adm. Reportes')
```

Figura 4-02: Script creación de roles predeterminados [A]

```
--Creación de la empresa proveedora del servicio

INSERT INTO dbo.EMPRESA VALUES ('0000' , 'Todos' , '' , '' , '' ,
'' , '' ,  'Todas las empresas' )

INSERT INTO dbo.EMPRESA VALUES ('0001', 'Soporte' , '' , '' , '' ,
'' , '' , 'Empresa Principal' )
```

Figura 4-03: Script creación de empresas predeterminadas [A]

```
--Creación del Usuarios del sistema

INSERT INTO dbo.USUARIO VALUES ( '0000' , '0000' , '0000' ,
'AFCZXKSU9A' , 'Todos' , '' )

INSERT INTO dbo.USUARIO VALUES ( '0001' , 'R001' , '0001' ,
'adm', 'Administrador', 'info@soporte.com' )

INSERT INTO dbo.USUARIO VALUES ( '0002' , 'R004' , '0001' ,
'clave', 'Adm. Empresas', 'info@soporte.com' )

INSERT INTO dbo.USUARIO VALUES ( '0003' , 'R005' , '0001' ,
'clave', 'Adm. Roles', 'info@soporte.com' )

INSERT INTO dbo.USUARIO VALUES ( '0004' , 'R006' , '0001' ,
'clave', 'Adm. Usuarios', 'info@soporte.com' )

INSERT INTO dbo.USUARIO VALUES ( '0005' , 'R007' , '0001' ,
'clave', 'Adm. Reportes', 'info@soporte.com' )
```

Figura 4-04: Script creación de usuarios Administradores [A]

Creación de las clases que interactuarán con la base de datos y que serán las entidades del sistema. Se utilizó la tecnología Ado .Net para solucionar el problema de impedancia que genera el modelo relacional frente a la programación orientada a objetos.

Generé una clase por cada tabla de la base de datos, de esta manera es posible acceder a los datos de forma ágil. La forma de acceder a la información es aplicando ***SQL to Entities***[15], de

[15] Es un lenguaje parecido a SQL que permite consultar los modelos conceptuales en Entity Framework. Los modelos conceptuales representan los datos como entidades y relaciones, y Entity SQL permite consultar estas entidades y relaciones en un formato similar a SQL.

esta manera se aplica sentencias SQL a las entidades/objetos y no directamente a la base de datos.

Creación de la pantalla para el ingreso al sistema y presentación de las opciones según corresponde el rol para garantizar que cada usuario tenga acceso únicamente a la información que le pertenece.

- **Retroalimentación**

Se sugirió incrementar roles individuales para cada una de las gestiones. Y se consideró importante la creación de la funcionalidad para cambiar la contraseña, y en caso de que el usuario no la recuerde se le enviará un correo electrónico de forma automática para evitar la molestia tanto al usuario al solicitar la contraseña como la del administrador al encontrarse con la tarea de restaurar contraseñas. Fue necesario realizar un ajuste para que en el momento del envió de la contraseña no se envíe la anterior, sino en su lugar se genere una nueva contraseña aleatoria para garantizar la seguridad.

Además, surgió la necesidad de crear un botón que garantice la finalización de la sesión de cada usuario con el objetivo de evitar que personas ajenas tengan acceso a información personal o delicada.

4.2.2. Iteración No.2

Creación del ***template***[16] de la aplicación y creación de la parte gráfica.

- **Procedimiento**

Creación del ***template*** del sistema mediante hojas de estilos. Está dividido en tres partes principales: cabecera, cuerpo, y pie de página.

En la parte izquierda de la cabecera se encuentra el logotipo de la aplicación, mientras que al lado derecho se puede visualizar el menú común para todos los usuarios. En el caso de que sea el administrador, se desplegarán las opciones de gestión y el usuario que se encuentra activo.

En el cuerpo se desplegará la información que corresponda a cada una de las páginas con su respectivo título en la parte superior.

Y, en la parte inferior se visualiza el pie de página, en el que se presenta los derechos de autor del sistema y en el cual se podrá colocar la información personalizada que desee cada empresa que utilice este sistema.

Es importante considerar que la información del usuario activo no estuvo considerada en el diseño pero en el transcurso del desarrollo fue necesario incrementarla.

[16] Diseño de una página web, el cual determina el layout, fuente, colores, imágenes, entre otros.

4.2.3. Iteración No.3

Creación de pantallas para la gestión con su respectiva funcionalidad.

- **Procedimiento**

Se estableció realizar tres pantallas iniciales: Gestión de roles, Gestión de empresas, y Gestión de usuarios; Mediante las cuales se tendrá acceso a los contratos y tareas según corresponda.

Se ha decidido colocar en una misma pantalla la funcionalidad de Crear, Actualizar, y Eliminar registros para que el usuario no tenga la necesidad de navegar por varias páginas al momento de realizar su trabajo. Por lo tanto, en una misma tabla se tendrá acceso al control total de la funcionalidad de cada módulo.

Creación de la navegación entre pantallas de gestión, las mismas que dieron origen a la Gestión de contratos que pertenecen a cada empresa, y a su vez, a la Gestión de tareas que pertenecen a cada contrato.

Esto facilitará al momento de desplegar las tareas de forma ordenada y así, el administrador del sistema tendrá fácil acceso a los datos.

- **Retroalimentación**

Luego de la revisión de esta iteración se consideró importante incrementar propiedades adicionales a los contratos: fecha de inicio, fecha de finalización, estado, y número o

referencia al documento escrito. Se consideró que no era necesario el manejo de horas en las fechas por lo cual fueron omitidas.

Adicionalmente en esta iteración fue necesario realizar ajustes en el script de la base de datos para que automáticamente genere los ***Primary Keys***[17] de la tabla Contrato y Tarea**.** Para aumentar esta funcionalidad fue necesario aumentar la propiedad Identity(0,1), en la cual el primer parámetro representa la semilla y el segundo la secuencia que debe seguir cada vez que se ingrese un nuevo registro.

También fue necesario incrementar botones que faciliten regresar de una pantalla a la anterior para facilitar la navegación.

Y por último fue necesario separar el ingreso de información a una pantalla diferente. Este cambio se lo realizó considerando que con el trascurso del tiempo la información puede incrementarse y seria innecesario visualizarla cuando únicamente se requiere ingresarla, lo que dio origen a incrementar las opciones de ingreso en el menú de gestión.

[17] Por su traducción “clave principal”, es una columna o combinación de columnas cuyos valores identifican de forma única una fila o registro de una tabla. La clave principal (s) tienen un valor único para cada registro o fila de la tabla.

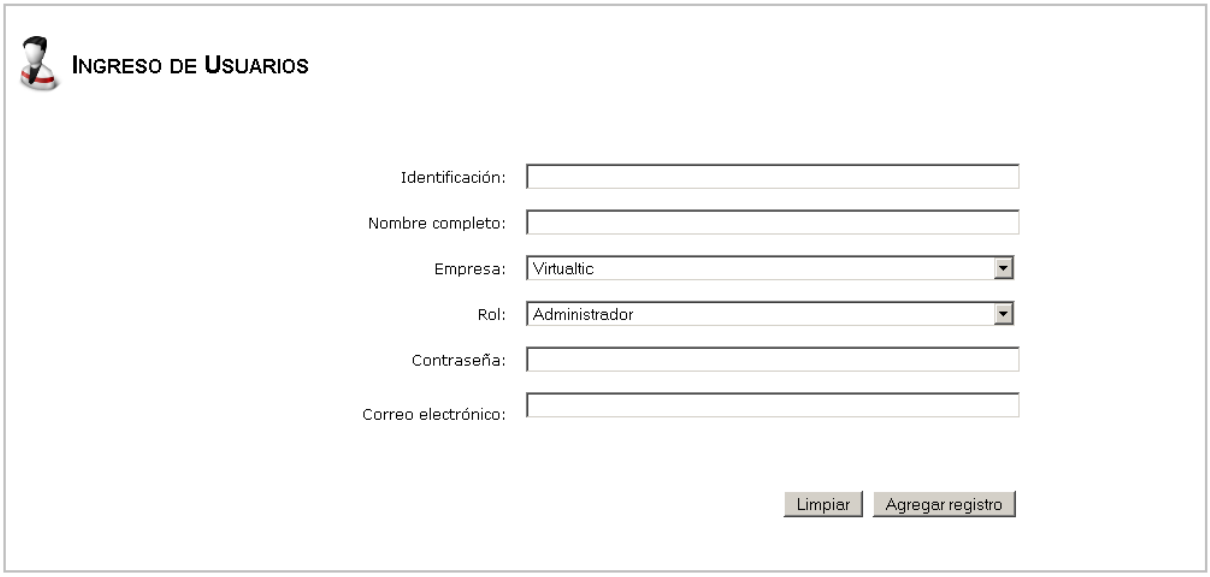

Figura 4-04: Pantalla tipo Ingreso [A]

Adicionalmente surgió la necesidad crear funciones de validación para garantizar el correcto funcionamiento del sistema. Incremente validaciones mediante expresiones regulares para verificar que no se ingrese información incorrecta.

Se determinó que es necesario que en las pantallas de gestión, al momento de modificar un registro, es necesario tener la posibilidad de cancelar la modificación para los cual procedí a retirar la validación **CauseValidation**[18] porque de esta manera a pesar de que existan errores se puede retornar sin interferir en los datos.

[18] Es la validación que desactiva los botones de una aplicación web mientras la información ingresada se corrigan los errores de validación.

Incremente validadores que informen al administrador cuando un registro ya existe y de esta manera poder evitar problemas de datos duplicados**.** Y, fue necesario colocar **ToolTips**[19] para presentar al usuario los respectivos mensajes informativos y de validación de forma sencilla y precisa. De esta manera el usuario puede identificar cada uno de los mensajes de error y entender la funcionalidad de los botones e iconos a utilizar.

Debido a que el número de datos contenidos en cada una de las tablas puede incrementarse significativamente con el transcurso del tiempo, se realizó la paginación en las tablas con lo cual en lo posible el usuario no tendrá la necesidad de utilizar el scrollbar.

Finalmente, ninguna tarea puede empezar con estado "Cerrado", dado que el sistema perdería toda su virtud de planificación. Y fue necesario cambiar los iconos de la aplicación por otros más intuitivos.

4.2.4. Iteración No.4

Creación de reportes de la aplicación, la cual será utilizada por todos los usuarios con los diferentes roles que interactuarán con el sistema.

[19] Es una herramienta de ayuda visual que funciona al situar o pulsar con el ratón sobre algún elemento gráfico, mostrando una ayuda adicional para informar al usuario de la finalidad del elemento sobre el que se encuentra. Los Tooltip son una variación de los globos de ayuda y es un complemento muy usado en programación, dado que proporcionan información adicional sin necesidad de que el usuario la solicite.

- **Procedimiento**

Para la creación de reportes se consideró importante la obtención de tareas de diversas maneras, dado que es la base del trabajo diario y puede ser útil al momento de planificación y verificación de procesos internos y externos si lo solicita el cliente.

La generación de reportes se la realiza previo ingreso de parámetros. Los parámetros involucrados son: empresas, contratos, usuarios, clase del servicio, tipo de servicio, y estado de tareas. De tal modo que se podrá obtener acceso a la información estrictamente necesaria, un ejemplo de un reporte podría ser "Todas las tareas que pertenecen a la empresa1, al contrato2, que fueron asignadas al técnico3, que se encuentran en estado pendiente y que son preventivas". Cabe recalcar que existen varias posibles combinaciones entre todos los parámetros mencionados, lo que da origen a gran variedad de reportes.

Figura 4-05: Sección parámetros para Reportes [A]

Una vez generado el reporte, puede ser exportado a un archivo Excel para que ser tabulado o realizar cálculos de ser necesario.

- **Retroalimentación**

A la finalización de esta iteración surgió la necesidad de incrementar la posibilidad de imprimir el reporte, para lo cual se incrementó la funcionalidad para exportar los reportes a formato ***PDF***[20] para que pueda ser impreso o enviado mediante correo electrónico.

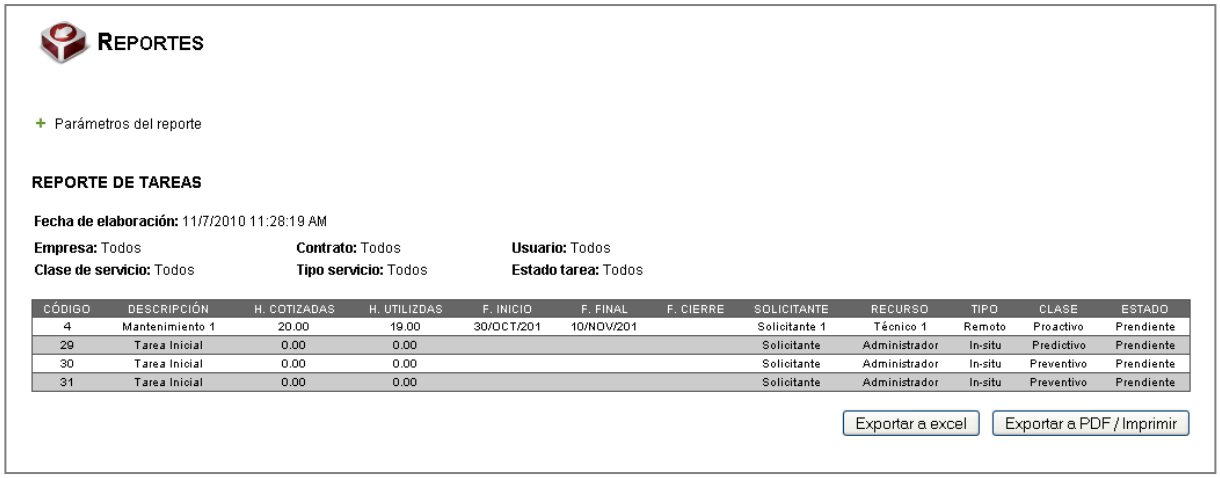

Figura 4-06: Reportes tipo del sistema [A]

[20] Acrónimo del inglés *portable document format*, formato de documento portátil) es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems

	A	B	C	D	E	F	G	H	I	J	K	L
1	CÓDIGO	DESCRIPCIÓN	H. COTIZADAS	H. UTILIZDAS	F. INICIO	F. FINAL	F. CIERRE	SOLICITANTE	RECURSO	TIPO	CLASE	ESTADO
2	4	Mantenimiento 1	20	19	30/OCT/201	10/NOV/201		Solicitante 1	Técnico 1	Remoto	Proactivo	Prendiente
3	29	Tarea Inicial	0	0				Solicitante	Administrador	In-situ	Predictivo	Prendiente
4	30	Tarea Inicial	0	0				Solicitante	Administrador	In-situ	Preventivo	Prendiente
5	31	Tarea Inicial	0	0				Solicitante	Administrador	In-situ	Preventivo	Prendiente

Figura 4-7: Reporte en Excel [A]

REPORTE DE TAREAS

Fecha de elaboracón:11/7/2010 11:28:19 AM

Empresa: Todos
Contrato: Todos
Responsable: Todos
Clase de servicio: Todos
Tipo de servicio: Todos
Estado tarea: Todos

CÓDIGO	DESCRIPCIÓN	H. COTIZADAS	H. UTILIZADAS	F. INICIO	F. FINAL	F. CIERRE	SOLICITANTE	RESPONSABLE	TIPO	CLASE	ESTADO
4	Mantenimiento 1	20.00	19.00	30/OCT/201	10/NOV/201		Solicitante 1	Técnico 1	Remoto	Proactivo	Prendiente
29	Tarea Inicial	0.00	0.00				Solicitante	Administrador	In-situ	Predictivo	Prendiente
30	Tarea Inicial	0.00	0.00				Solicitante	Administrador	In-situ	Preventivo	Prendiente
31	Tarea Inicial	0.00	0.00				Solicitante	Administrador	In-situ	Preventivo	Prendiente

Figura 4-8: Reporte en formato PDF [A]

En esta iteración se cumplió con el funcionamiento del administrador del sistema, sin embargo tenemos dos roles adicionales: Cliente y Técnico. Estos dos roles tendrán acceso a los reportes según les corresponda, por ejemplo el cliente tendrá la posibilidad de visualizar las tareas que le pertenecen a su empresa, mientras que el técnico deseará consultar las tareas que le han sido asignadas.

Es importante recalcar que el técnico no será la persona encargada de establecer cuando una tarea transite de un estado a otro, para esto, estará el administrador del sistema quien mediante

observación de cumplimiento de las etapas irá actualizando los estados para mantener la información íntegra y real.

En esta etapa fue necesario mediante variables de sesión restringir el acceso a las funciones administrativas, y fue factible mediante validación del rol asignado a cada usuario.

También fue necesario incrementar un campo para ingresar las observaciones de cada tarea, para lo cual cree una columna adicional en la base de datos. Y en los reportes aumente un mensaje de alerta que se desplegará cuando las tareas se encuentren retrasadas o hayan culminado atrasadas.

5. CAPITULO 5: Conclusiones y Recomendaciones

5.1. Conclusiones

1. Existen empresas, inclusive las que tienen como actividad la prestación de servicios informáticos y tecnológicos, que dedican exceso de tiempo en tratar de conocer y entender los aspectos tecnológicos que no forman parten se su propia línea del negocio, dejando de lado la inversión de ese tiempo para la mejora de sus procesos.

2. Los procesos de fabricación son operados como un sistema cerrado, mientras que los servicios operan como un sistema abierto con el impacto total de las variaciones en la demanda, lo cual también se transmite al sistema en el que interactúan.

3. Los servicios de soporte y mantenimiento informático que requieren capital alto requieren un monitoreo exhaustivo de los avances tecnológicos para mantenerse competitivos y a la vez requieren de que el gerente planifique la demanda para mantener el equipo activo; por ejemplo: las aerolíneas, los hoteles, la industria automotriz, entre otros. Mientras que cuando los servicios son intensivos en labor, es más importante centrarse en asuntos que involucran al personal que prestará los mismos.

4. Extreme Programming es una metodología de software enfocada para proyectos experimentales o prototipos dado que no brinda una documentación detallada como se requiere para proyectos grandes en los que se maneja altos niveles de transaccionalidad e interoperabilidad entre sistemas. Brinda la posibilidad de ir satisfaciendo las necesidades del usuario durante el transcurso de las iteraciones y presta agilidad al momento de codificar el sistema.

5. Las aplicaciones web son la tendencia actual dado que son accesibles desde cualquier lugar, las personas no tienen la necesidad de acudir a la oficina para desempeñar su trabajo diario, son accesibles desde varios dispositivos tecnológicos, y son atractivas e intuitivas de cara al usuario.

6. Los "planes honestos" son algo más que las estimaciones de la verdad; estos están estrechamente relacionados con una fecha de caducidad o ***deadline***, por lo tanto es indispensable ser honesto acerca de cuánto tiempo puede permanecer un plan vigente y cuando ya es necesario crear uno nuevo.

7. Un proceso ágil toma el proceso tradicional y lo convierte a noventa grados. Esto quiere decir que un sistema ágil permite a los administradores obtener un costo estimado por función en lugar de por actividad. Los clientes pueden tomar las decisiones difíciles sobre lo que puede ser dejado de hacer de una manera inteligente.

8. El mayor problema al desarrollar software son los cambios en las necesidades en el transcurso del proyecto. Los proyectos tienen requerimientos cambiantes, por lo que es imposible garantizar que no se susciten cambios en el camino, sin dejar de lado que siempre va a existir ámbitos en los requisitos que no pueden cambiar. Dado los antecedentes anteriormente mencionados cabe indicar la importancia de entender las necesidades del usuario para establecer correctamente los requerimientos que nos ayudarán al momento de presentar soluciones.

5.2. Recomendaciones

1. Es importante que el gerente de servicios adapte su lenguaje técnico al del cliente para mantener una comunicación efectiva de dos vías que garanticen la satisfacción de las necesidades.

2. Es indispensable seguir estrictamente la metodología aplicada en un proyecto para obtener resultados reales de lo propuesto, y de esta manera tener la posibilidad de medir los resultados obtenidos.

3. El modo de generar páginas dinámicas ha evolucionado, desde la utilización del CGI. Y al momento de optimizarlas es necesario diferenciar la funcionalidad que debe ir bajo un ***Server Side*** y las que deben ir en ***Client Side***, dado que de esta manera se agilita la navegación del usuario.

4. Para que esta disertación sea comprendida en su totalidad es importante que el lector se involucre con los capítulos teóricos y tenga conocimientos básicos de LINQ propuesto por Microsoft .Net. Siguiendo ordenadamente las iteraciones que se realizaron para la construcción del sistema, de esta manera podrá observar la totalidad de la funcionalidad.

5. Sugiero que la facultad de Ingeniería de Sistemas establezca los proyectos, que pertenecen a las materias en cuestión, de forma que se de soluciones a problemas reales dado que considero que de esta manera los estudiantes podemos involucrarnos con el tipo de actividad en la que nos desenvolveremos.

6. La Universidad debería generar mayor incentivo a los estudiantes para que estos se vean motivados a aplicar a becas de estudio e intercambio cultural. De tal modo que se pueda aplicar el intercambio de conocimiento.

ANEXOS

ANEXO 1.

Script de la base de datos

```
USE [Tesis]
GO
/****** Object:  Table [dbo].[CLASE]    Script Date: 02/12/2011 22:58:03
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[CLASE](
	[CLSECODIGO] [char](4) NOT NULL,
	[CLSEDESCRIPCION] [varchar](50) NULL,
	[CLSEICONO] [varchar](25) NULL,
 CONSTRAINT [PK_CLASE] PRIMARY KEY NONCLUSTERED
(
	[CLSECODIGO] ASC
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[EMPRESA]    Script Date: 02/12/2011 22:58:10
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[EMPRESA](
	[EMPRCODIGO] [char](4) NOT NULL,
	[EMPRNOMBRE] [text] NULL,
	[EMPRTELEFONO] [varchar](20) NULL,
	[EMPRMAIL] [varchar](100) NULL,
	[EMPRDIRECCION] [text] NULL,
	[EMPRCIUDAD] [varchar](60) NULL,
	[EMPRPAIS] [varchar](100) NULL,
	[EMPRACTIVIDAD] [text] NULL,
 CONSTRAINT [PK_EMPRESA] PRIMARY KEY NONCLUSTERED
(
	[EMPRCODIGO] ASC
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[ESTADO]    Script Date: 02/12/2011 22:58:11
******/
```

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[ESTADO](
	[ESTDCODIGO] [char](4) NOT NULL,
	[ESTDDESCRIPCION] [varchar](50) NULL,
	[ESTDICONO] [varchar](25) NULL,
 CONSTRAINT [PK_ESTADO] PRIMARY KEY NONCLUSTERED
(
	[ESTDCODIGO] ASC
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[ROL]    Script Date: 02/12/2011 22:58:12
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[ROL](
	[ROLCODIGO] [char](4) NOT NULL,
	[ROLDESCRIPCION] [varchar](50) NULL,
 CONSTRAINT [PK_ROL] PRIMARY KEY NONCLUSTERED
(
	[ROLCODIGO] ASC
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[TIPO]    Script Date: 02/12/2011 22:58:20
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TIPO](
	[TIPOCODIGO] [char](4) NOT NULL,
	[TIPODESCRIPCION] [varchar](50) NULL,
	[TIPOICONO] [varchar](25) NULL,
 CONSTRAINT [PK_TIPO] PRIMARY KEY NONCLUSTERED
(
	[TIPOCODIGO] ASC
```

```
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[TAREA]    Script Date: 02/12/2011 22:58:18
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[TAREA](
      [TREACODIGO] [bigint] IDENTITY(-1,1) NOT NULL,
      [TIPOCODIGO] [char](4) NOT NULL,
      [CLSECODIGO] [char](4) NOT NULL,
      [CNTRCODIGO] [bigint] NOT NULL,
      [USROIDENTIFICACION] [varchar](13) NOT NULL,
      [ESTDCODIGO] [char](4) NOT NULL,
      [TREADESCRIPCION] [text] NULL,
      [TREAHORASCOTIZADAS] [decimal](6, 2) NULL,
      [TREAHORASUTILIZADAS] [decimal](6, 2) NULL,
      [TREAFECHAINICIO] [varchar](11) NULL,
      [TREAFECHAFINAL] [varchar](11) NULL,
      [TREAFECHACIERRE] [varchar](11) NULL,
      [TREASOLICITANTE] [text] NULL,
      [TREAOBSERVACION] [text] NULL,
 CONSTRAINT [PK_TAREA] PRIMARY KEY NONCLUSTERED
(
      [TREACODIGO] ASC
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[USUARIO]    Script Date: 02/12/2011 22:58:23
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[USUARIO](
      [USROIDENTIFICACION] [varchar](13) NOT NULL,
      [ROLCODIGO] [char](4) NOT NULL,
      [EMPRCODIGO] [char](4) NOT NULL,
      [USROPASSWORD] [varchar](10) NOT NULL,
      [USRONOMBRE] [varchar](50) NOT NULL,
      [USROEMAIL] [varchar](100) NOT NULL,
 CONSTRAINT [PK_USUARIO] PRIMARY KEY NONCLUSTERED
(
```

```
	[USROIDENTIFICACION] ASC
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/****** Object:  Table [dbo].[CONTRATO]    Script Date: 02/12/2011 22:58:06
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[CONTRATO](
	[CNTRCODIGO] [bigint] IDENTITY(-1,1) NOT NULL,
	[EMPRCODIGO] [char](4) NOT NULL,
	[CNTRDESCRIPCION] [text] NULL,
	[CNTRHORASCONTRATADAS] [decimal](6, 2) NULL,
	[CNTRINICIO] [varchar](11) NULL,
	[CNTRFIN] [varchar](11) NULL,
	[CNTRESTADO] [char](1) NULL,
	[CNTRNUMERO] [nchar](15) NULL,
 CONSTRAINT [PK_CONTRATO] PRIMARY KEY NONCLUSTERED
(
	[CNTRCODIGO] ASC
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/****** Object:  ForeignKey [FK_CONTRATO_RELATIONS_EMPRESA]    Script Date:
02/12/2011 22:58:07 ******/
ALTER TABLE [dbo].[CONTRATO]  WITH CHECK ADD  CONSTRAINT
[FK_CONTRATO_RELATIONS_EMPRESA] FOREIGN KEY([EMPRCODIGO])
REFERENCES [dbo].[EMPRESA] ([EMPRCODIGO])
GO
ALTER TABLE [dbo].[CONTRATO] CHECK CONSTRAINT
[FK_CONTRATO_RELATIONS_EMPRESA]
GO
/****** Object:  ForeignKey [FK_TAREA_RELACION__CLASE]    Script Date:
02/12/2011 22:58:18 ******/
ALTER TABLE [dbo].[TAREA]  WITH CHECK ADD  CONSTRAINT
[FK_TAREA_RELACION__CLASE] FOREIGN KEY([CLSECODIGO])
REFERENCES [dbo].[CLASE] ([CLSECODIGO])
GO
ALTER TABLE [dbo].[TAREA] CHECK CONSTRAINT [FK_TAREA_RELACION__CLASE]
GO
/****** Object:  ForeignKey [FK_TAREA_RELACION__CONTRATO]    Script Date:
02/12/2011 22:58:18 ******/
ALTER TABLE [dbo].[TAREA]  WITH CHECK ADD  CONSTRAINT
[FK_TAREA_RELACION__CONTRATO] FOREIGN KEY([CNTRCODIGO])
REFERENCES [dbo].[CONTRATO] ([CNTRCODIGO])
```

```
GO
ALTER TABLE [dbo].[TAREA] CHECK CONSTRAINT [FK_TAREA_RELACION__CONTRATO]
GO
/****** Object:  ForeignKey [FK_TAREA_RELACION__ESTADO]    Script Date:
02/12/2011 22:58:18 ******/
ALTER TABLE [dbo].[TAREA]  WITH CHECK ADD  CONSTRAINT
[FK_TAREA_RELACION__ESTADO] FOREIGN KEY([ESTDCODIGO])
REFERENCES [dbo].[ESTADO] ([ESTDCODIGO])
GO
ALTER TABLE [dbo].[TAREA] CHECK CONSTRAINT [FK_TAREA_RELACION__ESTADO]
GO
/****** Object:  ForeignKey [FK_TAREA_RELACION__TIPO]    Script Date:
02/12/2011 22:58:18 ******/
ALTER TABLE [dbo].[TAREA]  WITH CHECK ADD  CONSTRAINT
[FK_TAREA_RELACION__TIPO] FOREIGN KEY([TIPOCODIGO])
REFERENCES [dbo].[TIPO] ([TIPOCODIGO])
GO
ALTER TABLE [dbo].[TAREA] CHECK CONSTRAINT [FK_TAREA_RELACION__TIPO]
GO
/****** Object:  ForeignKey [FK_TAREA_RELACION__USUARIO]    Script Date:
02/12/2011 22:58:19 ******/
ALTER TABLE [dbo].[TAREA]  WITH CHECK ADD  CONSTRAINT
[FK_TAREA_RELACION__USUARIO] FOREIGN KEY([USROIDENTIFICACION])
REFERENCES [dbo].[USUARIO] ([USROIDENTIFICACION])
GO
ALTER TABLE [dbo].[TAREA] CHECK CONSTRAINT [FK_TAREA_RELACION__USUARIO]
GO
/****** Object:  ForeignKey [FK_USUARIO_RELACION__EMPRESA]    Script Date:
02/12/2011 22:58:23 ******/
ALTER TABLE [dbo].[USUARIO]  WITH CHECK ADD  CONSTRAINT
[FK_USUARIO_RELACION__EMPRESA] FOREIGN KEY([EMPRCODIGO])
REFERENCES [dbo].[EMPRESA] ([EMPRCODIGO])
GO
ALTER TABLE [dbo].[USUARIO] CHECK CONSTRAINT [FK_USUARIO_RELACION__EMPRESA]
GO
/****** Object:  ForeignKey [FK_USUARIO_RELACION__ROL]    Script Date:
02/12/2011 22:58:23 ******/
ALTER TABLE [dbo].[USUARIO]  WITH CHECK ADD  CONSTRAINT
[FK_USUARIO_RELACION__ROL] FOREIGN KEY([ROLCODIGO])
REFERENCES [dbo].[ROL] ([ROLCODIGO])
GO
ALTER TABLE [dbo].[USUARIO] CHECK CONSTRAINT [FK_USUARIO_RELACION__ROL]
GO
```

ANEXO 2.

Script de inicialización de la base de datos

```
--Creación de roles del sistema
INSERT INTO dbo.ROL VALUES ('0000' , 'Todos'         )
INSERT INTO dbo.ROL VALUES ('R001' , 'Administrador')
INSERT INTO dbo.ROL VALUES ('R002' , 'Cliente'       )
INSERT INTO dbo.ROL VALUES ('R003' , 'Técnico'       )
INSERT INTO dbo.ROL VALUES ('R004' , 'Adm. Empresas')
INSERT INTO dbo.ROL VALUES ('R005' , 'Adm. Roles'    )
INSERT INTO dbo.ROL VALUES ('R006' , 'Adm. Usuarios')
INSERT INTO dbo.ROL VALUES ('R007' , 'Adm. Reportes')


--Creación de la empresa proveedora del servicio
INSERT INTO dbo.EMPRESA VALUES ('0000' , 'Todos' , '' , '' , '' , ''
, '' ,  'Todas las empresas' )
INSERT INTO dbo.EMPRESA VALUES ('0001', 'Soporte' , '' , '' , '' , ''
, '' , 'Empresa Principal' )


--Creación del Usuario Administrador del sistema
INSERT INTO dbo.USUARIO VALUES ( '0000' , '0000' , '0000' ,
'AFCZXKSU9A' , 'Todos' , '' )
INSERT INTO dbo.USUARIO VALUES ( '0001' , 'R001' , '0001' , 'adm',
'Administrador', 'info@soporte.com' )
INSERT INTO dbo.USUARIO VALUES ( '0002' , 'R004' , '0001' , 'clave',
'Adm. Empresas', 'info@soporte.com' )
INSERT INTO dbo.USUARIO VALUES ( '0003' , 'R005' , '0001' , 'clave',
'Adm. Roles', 'info@soporte.com' )
INSERT INTO dbo.USUARIO VALUES ( '0004' , 'R006' , '0001' , 'clave',
'Adm. Usuarios', 'info@soporte.com' )
INSERT INTO dbo.USUARIO VALUES ( '0005' , 'R007' , '0001' , 'clave',
'Adm. Reportes', 'info@soporte.com' )
```

BILIOGRAFIA

[A] Paolo Aguilar, *Desarrollo De Un Sistema De Administración De Tareas De Soporte Y Mantenimiento Informático. 2010.*

[B] Roger Pressman. *Ingeniería del Software: Un Enfoque Practico*. McGraw-Hill. 2006

[C] Stephen R. Schach. *Ingeniería de Software Clásica y Orientada a Objetos*. McGraw-Hill. 2006

[D] Stanton William, Etzel Michael y Walker Bruce. *Fundamentos de Marketing*, 13va. Edición, Mc Graw Hill, 2004.

[E] Sandhusen L. Richard. *Mercadotecnia*, Primera Edición, Editorial Continental, 2002.

[F] Lamb Charles, Hair Joseph y McDaniel Carl. *Marketing*, Sexta Edición, International Thomson Editores, 2006.

[G] Kotler Philip, Bloom Paul y Hayes Thomas, *El marketing de Servicios Profesionales*, Editorial Paidós SAICF, 2004.

[1] http://www.cin.ufpe.br/~redis/intranet/bibliography/software_architecture/maier-147101.pdf

[2] http://www.extremeprogramming.org/

[3] *MarketingPower.com*, de la *American Marketing Association*, Sección Dictionary of Marketing Terms, Obtenido el 2 de Octubre del 2010.