

**PRÁCTICA EMPRESARIAL PARA ISL - INGENIEROS DE SISTEMAS LTDA. -
APOYO A LA IMPLEMENTACION DE NUEVAS METODOLOGIAS Y
HERRAMIENTAS TECNOLOGICAS PARA EL DISEÑO Y DESARRLLO DE
APLICACIONES EN AMBIENTE WEB.**

JOSÉ VICENTE SILVA RIVERA

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2007

**PRÁCTICA EMPRESARIAL PARA ISL - INGENIEROS DE SISTEMAS LTDA. -
APOYO A LA IMPLEMENTACION DE NUEVAS METODOLOGIAS Y
HERRAMIENTAS TECNOLOGICAS PARA EL DISEÑO Y DESARRLLO DE
APLICACIONES EN AMBIENTE WEB.**

JOSÉ VICENTE SILVA RIVERA.

Trabajo de grado para obtener el titulo de Ingeniero de Sistemas

**Tutor: Ing. Luís Antonio Flórez Flórez
Gerente General
ISL. INGENIEROS DE SISTEMAS LTDA.**

**Director: Ing. Leonel Parra Pinilla
Escuela de Sistemas UIS**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2007

“el día en que ya no pueda levantarme y seguir luchando, ese día... ese día será mi muerte”

Dedicatoria

Dedicado a Dios mi compañero y mi esperanza en todo momento de este camino llamado vida, y a toda mi familia, en especial a mi madre Lilia, mi padre Trino José, mi hermana Julia Andrea, y mi tía Teresa, todos ellos han sido mi motor, me brindaron su apoyo incondicional y su fortaleza en los momentos difíciles para no declinar y rendirme, su voz de aliento me permitió levantarme una y otra vez para seguir luchando por esta causa; se que están orgullosos de ver cumplida esta meta, que de no ser por ellos no habría sido posible. Doy gracias a Dios por tener a mi lado personas tan maravillosas y espero

AGRADECIMIENTOS

- ISL Ingenieros de sistemas Ltda. En especial al ingeniero Luis Antonio Flórez y la ingeniera Maria Teresa Castro, por brindarme la confianza para realizar este trabajo y permitirme representar a su organización, gracias por su apoyo sobre todo en los momentos difíciles y por su esfuerzo para lograr la realización de esta práctica
- Al grupo de trabajo de ISL que aunque fue cambiando durante todos los momentos, siempre existió la camaradería que caracteriza un gran equipo
- La Escuela de Ingeniería de Sistemas de la UIS al profesor Fernando Ruiz director de escuela, a Maria Cecilia la secretaria, y todo el personal que la conforma por su gran y sincera colaboración que permitieron la realización y culminación de este objetivo
- Al profesor Leonel Parra Pinilla mi director de proyecto, quien no solamente se limitó a mostrarme los lineamientos a seguir, sino también por su gran profesionalismo y calidad humana que lo hicieron comprometerse con mi trabajo y defenderlo con gran convicción
- A mis calificadores, profesor José Cárcamo Sepúlveda y profesor Fernando Antonio Rojas, por valorar mi trabajo durante estos seis meses y entender el esfuerzo realizado y los resultados obtenidos
- A la ingeniera Yenny Adriana Guevara, gracias a su organización y disciplina que me forzaban más a comprometerme con el trabajo, sin ella tal vez esto todavía

no hubiera terminado, una compañera de labores que logro colarse en el círculo bueno ese y existas en su vida profesional.

- A mis José Luis, Álvaro, Juan Manuel, Johan, Luis Genaro, Nelson, Jairo, Juan Camilo, Fabián, Beto, Cesar, amigos que con su apoyo y compañía me brindaron el entusiasmo y la energía para seguir adelante, espero seguir contando con estas maravillosas personas
- A esas mujeres encantadoras Viviana, Gissela, Lorena, Nercy, Melba, Yuly, Faisan Jasibe, Maria Fernanda, Astrid, Johana, Angelica, Ingrid, Ana Rosa, que hicieron parte de mi vida en algun momento y que con su sonrisa lograban cambiarme el animo, alegrame el momento y llenar de luz un dia opaco
- Este agradecimiento esta vacío, como vacante ha estado este puesto durante mucho tiempo en mi vida, se que la dueña esta ahí y donde este envió su energía y fuerza para que lograra mis metas.
- A mi primo Orlando casi mi hermano mayor, no alcanzan las palabras para expresar las gracias
- No me alcanza la hoja para nombrar a todas esas personas, familiares amigos y compañeros que de alguna u otra forma intervinieron para lograr esta meta que termina, a todos ellos mi infinita gratitud.

CONTENIDO

	pág.
INTRODUCCION	14
1. DESCRIPCION DE LA PRÁCTICA	16
1.1 DESCRIPCION DE LA EMPRESA	16
1.1.1 Nombre De La Empresa.	16
1.1.2 Misión.	16
1.1.3 Visión.	16
1.1.4 Valores.	17
1.1.5 Descripción Y Alcances Del Sistema De Gestión De Calidad.	17
1.1.6 Alcance De La Certificación.	18
1.1.7 Estructura Organizacional.	19
1.1.8 Responsabilidades a Cargo.	19
1.2 DESCRIPCION DEL PROYECTO	21
1.2.1 Planteamiento del problema.	21
1.2.2 Objetivos.	22
1.2.3 Justificación.	23
1.2.4 Cronograma de actividades.	26
2. MARCO TEORICO	28
2.1 FUNDAMENTACION METODOLOGICA	28

2.1.1 Metodologías de desarrollo de software.	28
2.1.2 Metodologías Utilizadas	57
2.2 APLICACIONES DISTRIBUIDAS	59
2.2.1 Filosofía y ventajas.	60
2.3 APLICACIONES WEB	63
2.3.1 funcionamiento	64
2.3.2 Arquitectura.	66
2.4 TECNOLOGIA .NET	69
2.4.1 ¿Que es .NET?.	69
2.4.2 Orígenes e Ideología.	69
2.4.3 Aplicaciones Web con ASP .NET de Visual Studio.	72
3. DESARROLLO DE LA PRÁCTICA	80
3.1 DESCRIPCION DE LA APLICACIÓN	81
3.1.1 Modulo de inventario.	81
3.1.2 Modulo de Cartera.	81
3.1.3 Modulo de facturación.	81
3.1.4 Modulo de Comisiones Nomina.	81
3.2 ANALISIS Y DISEÑO DE LA APLICACIÓN	82
3.2.1 Estudio y refinación del los requerimientos del proyecto.	83
3.2.2 Apoyo a la definición y documentación del diseño de la aplicación.	84
3.3 CONSTRUCCION DE LA APLICACIÓN	88
3.3.1 Diseño y construcción del primer prototipo.	89

3.3.2 Diseño y construcción del Segundo prototipo.	91
3.3.3 Diseño y Construcción del Tercer y Cuarto Prototipo.	92
4. DESCRIPCION DE OTRAS ACTIVIDADES Y RESPONSABILIDADES A CARGO	94
4.1 ACTIVIDADES DE SOPORTE	94
4.1.1 Apoyo al soporte a usuarios de los sistemas existentes dentro de la organización.	94
4.1.2 Instalación y configuración de hardware y software.	96
4.1.3 Mantenimiento de los diferentes sistemas de la empresa.	96
4.1.4 Apoyo al desarrollo software de aplicaciones existentes.	97
4.2 ACTIVIDADES DE HELP DESK	98
4.2.1 Recepción de solicitudes y problemas reportados por los usuarios.	98
4.2.2 Diagnostico preeliminar de problemas y registro de seguimiento a la solución de los mismos.	98
4.3 ACTIVIDADES DE CALIDAD	99
4.3.1 Apoyo al mejoramiento continuo de la organización desde los procesos a cargo.	99
5. CONCLUSIONES Y RECOMENDACIONES	100
5.1 CONCLUSIONES	100
5.2 RECOMENDACIONES	101
BIBLIOGRAFIA	102

LISTA DE TABLAS

pág.

Tabla 1. Cronograma de actividades

26

LISTA DE FIGURAS

	pág.
Figura 1. Estructura Organizacional de ISL	19
Figura 2. Diagrama de modelo en cascada	29
Figura 3. Modelo de Prototipado Evolutivo	33
Figura 4. Diagrama del Modelo en espiral	43
Figura 5. Diagrama de modelo XP	47
Figura 6. Estructura de RUP	54
Figura 7. Diagrama de Aplicaciones distribuidas	63
Figura 8. Funcionamiento de Aplicaciones Web	65
Figura 9. Representación del modelo tres capas	68
Figura 10. Contexto del.Net Framework	74

RESUMEN

TÍTULO: PRÁCTICA EMPRESARIAL PARA ISL - INGENIEROS DE SISTEMAS LTDA. - APOYO A LA IMPLEMENTACION DE NUEVAS METODOLOGIAS Y HERRAMIENTAS TECNOLOGICAS PARA EL DISEÑO Y DESARROLLO DE APLICACIONES EN AMBIENTE WEB.*

AUTOR: SILVA RIVERA, José Vicente **

PALABRAS CLAVES: Metodologías de desarrollo software, Visual Studio .NET, Sistema de Información Web, UML.

DESCRIPCION:

ISL Ingenieros de Sistemas Ltda. Organización dedicada al desarrollo de soluciones informáticas a la medida y a la prestación de servicios de administración y contratación de TI (*Tecnologías de la Información*), y que se ha caracterizado por estar a la vanguardia en el uso de metodologías y herramientas tecnológicas, para brindar a sus clientes soluciones novedosas de gran calidad que le han permitido obtener y mantener durante cuatro años certificados sus procesos con la norma ISO 9001:2000 por parte de ICONTEC. Lo anterior los motivó a buscar estudiantes con sus conocimientos frescos, para asumir los nuevos retos informáticos que nos presenta la ingeniería de sistemas actualmente.

El desarrollo de la practica giró en torno a tres pilares fundamentales: en primer lugar, apoyar el acoplamiento de los procesos y procedimientos actuales, con las nuevas metodologías de desarrollo software y herramientas tecnológicas enfocadas al ambiente Web; en segundo lugar, utilizar los conocimientos adquiridos en ingeniería de software para colaborar en la construcción de aplicaciones para el nuevo proyecto que consiste en un sistema de información orientado a Web, para el manejo de la venta y ocupación de servicios funerarios para un parque cementerio; por ultimo se realizaron tareas adicionales como mantenimiento de software para aplicaciones existentes, atención y capacitación de usuarios, servicio al cliente y Help Desk, contempladas dentro del manual de responsabilidades del cargo asignado, y que permitieron crecimiento profesional del estudiante en práctica. La práctica empresarial permite desarrollar en el estudiante, experiencia y habilidades en el entorno laboral que le permitirán desempeñar mejor sus funciones de ingeniería. Además, el uso de tecnologías .Net facilita el desarrollo de aplicaciones Web de alta calidad.

* Proyecto de Grado en la modalidad de Práctica Empresarial.

** Escuela de Ingeniería de Sistemas e Informática UIS. Director: Ing. Leonel Parra Pinilla.

ABSTRACT

TITLE: BUSINESS PRACTICE FOR ISL - INGENIEROS DE SISTEMAS LTDA. – SUPPORT TO THE IMPLEMENTATION OF NEW METHODOLOGY AND TECHNOLOGICAL TOOLS FOR DESIGNING AND DEVELOPMENT OF WEB APPLICATIONS. †

AUTHOR: SILVA RIVERA, José Vicente **

KEY WORDS: Software development methodologies, Visual Studio .NET, Web Information System, UML.

DESCRIPTION

ISL Ingenieros de Sistemas Ltda. is an organization dedicated to the development of System solutions, administration of provision of services and contracting IT (*Information Technology*). This company has been characterized to be in the vanguard of using methodologies and technological tools in order to offer ingenious solutions of high quality to its clients, which has allowed to obtain and to maintain during four years the ISO 9001:2000 certification, given by ICONTEC. Thus, the organization searched for students with fresh knowledge in order to assume new computing challenges of our current System Engineering.

Engineering practice was based on three fundamental pillars: first of all, to help to the connection of processes and actual procedures with new methodologies of software development and technological tools based on Web environment; secondly, to use acquired knowledge on Software Engineering to contribute in constructing applications to the new project which consists on a Web information system in order to sale funerary services for a cemetery; and finally, additional assignments were developed as software maintenance for existing applications, user support, and Help Desk, provided in the manual of responsibilities of the assigned charge, which permitted the professional growing of the student in practice.

The business practice allows developing expertise and skills to the student in the laboral environment, which will allow him to exercise its engineer functions. Besides, the use of .Net technologies makes easier to develop high quality Web applications

† Proyecto de Grado en la modalidad de Práctica Empresarial.

** Escuela de Ingeniería de Sistemas e Informática UIS. Director: Ing. Leonel Parra Pinilla.

INTRODUCCION

La modalidad de práctica empresarial, es una excelente oportunidad para el estudiante de enfrentar los conocimientos adquiridos durante sus años de estudio, a la realidad representada en el ambiente laboral, en donde se probarán no solo sus cualidades intelectuales sino que además se medirán sus habilidades para reaccionar ante los continuos cambios del medio, la facilidad para relacionarse con otros miembros del equipo de trabajo, la relación con los clientes, entre otras muchas vivencias que le permitirán al alumno adquirir la experiencia necesaria a la hora de enfrentar el mercado laboral y empresarial permanentemente.

Actualmente la creación de nuevas tecnologías crece vertiginosamente, así como la aparición de nuevas y mejores metodologías para el desarrollo de proyectos informáticos, además del auge de las aplicaciones orientadas a la Web, ha llevado a las empresas desarrolladoras de software a modernizar sus perspectivas para permanecer en el mercado.

ISL Ingenieros de Sistemas Ltda. no es ajena a estos cambios, y por tal motivo planea incursionar en la apropiación y utilización de tecnologías .NET y actualizar sus metodologías de producción de software, para tal fin requiere de material humano, con capacidades, visión emprendedora y conocimientos frescos, que les permitan alcanzar sus objetivos, encontrando en el ofrecimiento de practicas empresariales para estudiantes que están finalizando sus estudios un nicho de recursos humanos de gran calidad técnica y de excelente perfil profesional, como son los estudiantes de la Universidad Industrial de Santander reconocidos ampliamente a nivel regional y nacional.

El presente documento realiza una recopilación teórica y practica de los eventos y actividades realizadas durante los seis meses de desarrollo de la práctica empresarial, en donde se resume todas las experiencias adquiridas por parte del estudiante en practica, así como las recomendaciones conclusiones obtenidas tras la realización de la misma.

1. DESCRIPCION DE LA PRÁCTICA

En el presente capitulo se describen aspectos generales de la organización en la que se desarrolló la practica empresarial, tales como la razón social, reseña histórica y campo de acción, axial como los principales lineamientos de su sistema de gestión de calidad, además se presenta una descripción de los objetivos y alcances del proyecto, a la vez que cuantifica su impacto y viabilidad

1.1 DESCRIPCION DE LA EMPRESA

1.1.1 Nombre De La Empresa.

Razón Social: **ISL – Ingenieros de Sistemas Ltda.**

Tipo de Organización: Sociedad Limitada.

Fecha de Fundación: 3 de marzo de 1992.

Domicilio: Calle 35 No. 17 –77 Of. 401 Edificio. Bancoquia.

Ciudad: Bucaramanga

1.1.2 Misión. Ofrecer servicios informáticos oportunos y de calidad, mediante el análisis, la creatividad, la innovación y la optimización de métodos y herramientas que contribuyan al crecimiento y desarrollo de nuestros clientes.

1.1.3 Visión. “Ser reconocidos a nivel regional como la organización de base tecnológica líder en innovación y conocimiento de servicios informáticos integrales, con productos ampliamente posicionados y una sólida estructura organizacional, garantizando el desarrollo personal y profesional del recurso humano y estando a

la vanguardia en el monitoreo de los cambios y transformaciones que se presenten para dar respuesta y valor agregado a nuestros clientes”

1.1.4 Valores.

Trabajar en equipo para lograr los objetivos que se propongan en un ambiente de cordialidad, confianza y respeto mutuo.

Facilitar la aplicación de las capacidades de cada individuo para afrontar los retos que se presenten en cumplimiento de sus objetivos.

Buscar soluciones creativas e innovadoras para el cumplimiento de los objetivos planteados.

Dar libertad para expresar sugerencias y reclamos. La administración de ISL evalúa y responde en forma justa.

Promover el desarrollo personal permanente en forma integral, tanto profesional como humana.

Proveer servicios oportunos de la más alta calidad que sean consistentes con las necesidades y preferencias de los clientes, logrando la satisfacción de sus necesidades.

1.1.5 Descripción Y Alcances Del Sistema De Gestión De Calidad. ISL Ingenieros de sistemas Ltda. Se encuentra certificada desde hace cuatro años bajo la norma ISO 9001:2000, que certifica la calidad de sus procesos, recientemente le fue renovada su certificación por los próximos tres años, y le fue

ampliado el alcance, lo cual refleja el posicionamiento y liderazgo de esta organización frente a otras empresas de su genero.

1.1.6 Alcance De La Certificación. “Diseño, desarrollo, implantación y mantenimiento de software para la gestión administrativa y del negocio. Prestación de servicios de: gerencia e interventoría de proyectos administrativos y técnicos de IT, administración de redes, soporte a usuarios, administración y soporte a infraestructura informática, suministro de hardware y software.”

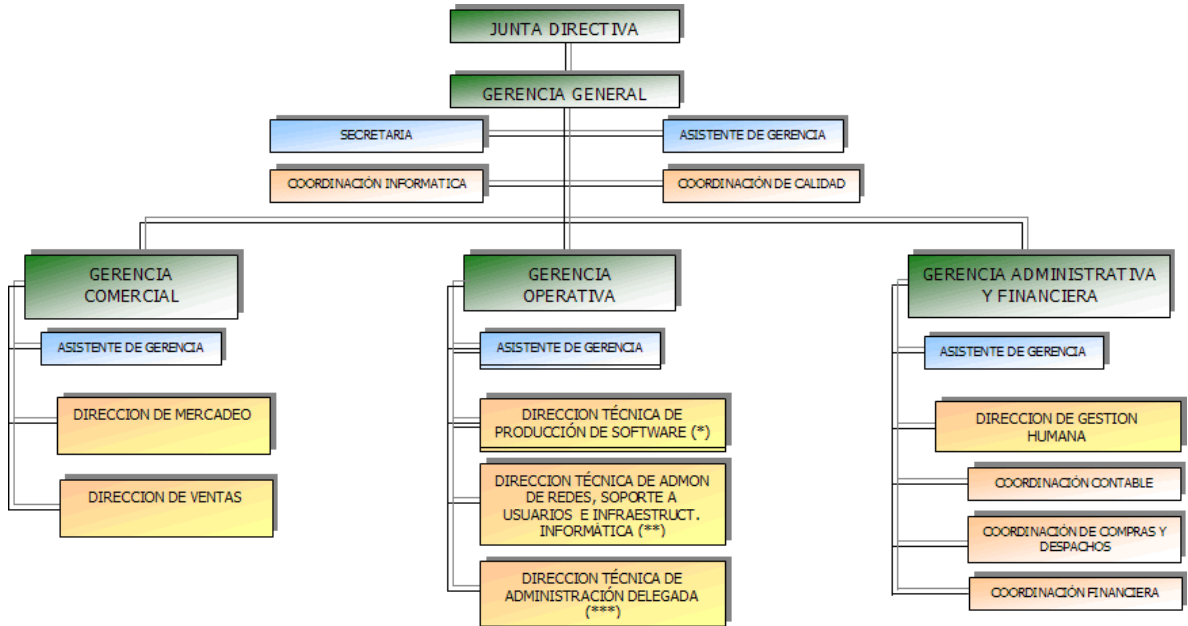
- **Política De Calidad.** Lograr la satisfacción de las necesidades informáticas del cliente identificando oportunidades que permitan mejorar la calidad de nuestros servicios, procesos y sistema de gestión en general.

- **Objetivos De Calidad.**

- Alcanzar un nivel del 80% de satisfacción del cliente
- Disminuir en un 10% el número de Quejas y reclamos
- Mejorar la eficiencia del sistema de gestión de calidad en un 5%
- Alcanzar un nivel cumplimiento del 90% en el tiempo de producción de software

1.1.7 Estructura Organizacional.

Figura 1. Estructura Organizacional de ISL



1.1.8 Responsabilidades a Cargo.

1. Brindar asesoría y soporte técnico a los usuarios de los equipos de micro computación para la resolución de dudas y problemas relacionados con la operación de los equipos, aplicaciones y programas instalados.
2. Servir de soporte básico a los sistemas de información, entendiéndose por esto la solución a problemas de impresión, manipulación de archivos, estrategias de respaldo y recuperación, incorrecta instalación y demás actividades directamente ligadas con la informática.

3. Instalar, configurar y, de ser necesario, reconfigurar equipos de computo, impresoras, graficadores, computadores, programas, unidades de equipo, etc.
4. Diagnosticar e identificar errores de mal funcionamiento de software y hardware instalados en la infraestructura informática.
5. Identificar tendencias y tomar las acciones conducentes a remediar globalmente aquellos problemas que repetitivamente son reportados y/o requeridos por su trabajo.
6. Brindar apoyo en las labores relacionadas con instalación y traslado de equipos y periféricos.
7. Realizar modificaciones menores a los archivos elaborados con las aplicaciones de software básico y realización de nuevos desarrollos.
8. Coordinar las labores de mantenimiento correctivo y preventivo de los microcomputadores y sus periféricos.

Help Desk:

9. Actuar como punto de contacto para la recepción y registro de las solicitudes o problemas reportados por los usuarios.
10. Diagnosticar las solicitudes, brindando asesoría y soporte técnico a aquellas que pueda resolver.
11. Registrar toda la información necesaria para el seguimiento y optimización del servicio.

Calidad

12. Controlar los documentos del sistema de calidad que estén bajo su responsabilidad participando activamente en la revisión de los mismos.

13. Generar y ejercer un control sobre los registros del sistema de gestión de la calidad que estén contemplados dentro de sus procesos y responsabilidades, asegurando su identificación, almacenamiento y protección.

14. Comunicar y reportar cualquier eventualidad o situación que potencialmente pueda generar una no conformidad, siguiendo el procedimiento de acciones preventivas.

15. Aplicar las acciones necesarias para corregir las no conformidades que se puedan presentar, generando las respectivas acciones correctivas y planes de mejora que eviten posteriores ocurrencias.³

1.2 DESCRIPCION DEL PROYECTO

1.2.1 Planteamiento del problema. ISL actualmente esta incursionando en el desarrollo de soluciones informáticas utilizando plataformas orientadas a Web como las tecnologías .Net, el principal inconveniente es que no se cuenta con personal experimentado en este tipo de tecnologías, y no se pueden detener los proyectos que se encuentran en ejecución hasta encontrar personal capacitado en estas áreas.

³ Tomado del documento **GAGR01 MANUAL DE RESPONSABILIDADES** (Documento Interno)

Es entonces idóneo para la empresa brindar un espacio para el desarrollo de investigación y estudio de estas tecnologías a estudiantes que se encuentren en la fase final de su carrera de ingeniería, para que utilizando estas tecnologías apliquen los conocimientos adquiridos y de paso permitan el avance de los proyectos represados

1.2.2 Objetivos.

- **Objetivo General.** Brindar Apoyo En Los Distintos Procesos De Diseño Y Desarrollo Software Al Interior De La Empresa, Para Lograr El Acoplamiento De La Misma Con La Tecnologia Basada En Ambiente Web

- **Objetivos Específicos.**

- Conocer, entender y aplicar los lineamientos básicos de calidad implementados por la empresa en el diseño y desarrollo de sus aplicaciones.

- Estudiar y presentar a la empresa las ventajas de la utilización de la metodología de "Diseño orientado a objetos con UML" y el impacto positivo que tienen estas dentro de los procesos y procedimientos de producción de software

- Apoyar la realización del diseño de los módulos inventario y comisiones del "sistema de información y control de productos tradicionales para un parque cementerio" en desarrollo, utilizando la metodología de "Diseño Orientado a Objetos con UML"

- Apoyar a la empresa en su política de implantación de la tecnología Visual Studio .NET 2005 así como la utilización de esta herramienta para desarrollo de

aplicaciones en ambiente Web con diversos motores de bases de datos como SQL u ORACLE entre otros.

- Basado en el análisis y la documentación de requerimientos existente, brindar apoyo en la construcción e implementación de los módulos de inventario y comisiones del “sistema de información y control de productos tradicionales para un parque cementerio” que actualmente se esta desarrollando en la empresa.
- Participar activamente en los procesos, labores y actividades de soporte a usuarios tanto del proyecto de desarrollo actual como en las demás aplicaciones de la organización.

1.2.3 Justificación. El mercado laboral de la profesión es muy competido a todo nivel; tanto los profesionales recién egresados como aquellos con muchos años de experiencia tienen dificultades a la hora de ver aprobada su hoja de vida en las diferentes empresas a las que se postulan.

Antes de utilizar las habilidades adquiridas en la universidad para dar vida a una nueva empresa, es prudente incursionar en el ambiente laboral y así obtener el conocimiento necesario del ambiente real laboral que solo la experiencia puede otorgar.

ISL cuenta con experiencia y proyección en distintos campos de la ingeniería como son: diseño, desarrollo, implementación y mantenimiento de software, diseño e instalación de redes y soporte y mantenimiento de plataformas e infraestructura informática; las cuales deben estar al día con las metodologías y herramientas tecnológicas de punta que permitan a la empresa permanecer a la vanguardia.

En los últimos años la tecnología informática ha venido creciendo vertiginosamente, y es necesario para cualquier empresa estar al día con las últimas tendencias tecnológicas para así lograr mantenerse en un mercado que a su vez es muy competido; para tal fin ISL se encuentra en un proceso de adaptación a nuevas herramientas como VISUAL STUDIO .NET 2005 que le permitirán desarrollar aplicaciones orientadas a ambiente Web y por lo tanto necesita personal capacitado que le pueda brindar apoyo en la estandarización en los procesos y metodologías para el uso de estos nuevos programas.

- **Impacto.** La nueva tendencia a la globalización de las distintas empresas hace que en estas surja la necesidad de nuevos y sofisticados productos, que no solo satisfagan las exigencias de informáticas de la misma, sino que además le permitan facilitar el proceso de ampliación y crecimiento del negocio.

El hecho de que ISL incurriera en el desarrollo de aplicaciones a la medida orientadas a entorno Web, genera un gran impacto tecnológico en la empresa, pues tiene que adaptar sus prácticas, procesos y metodologías actuales para que se acoplen perfectamente con las nuevas tendencias de soluciones informáticas requeridas por sus clientes, conservando por supuesto los altos estándares de calidad que la caracterizan, y que le han permitido mantener el certificado de calidad ISO 9001-2000 durante tres años consecutivos.

- **Viabilidad.** Las posibilidades tecnológicas para realizar el presente proyecto son bastante amplias, pues cuando se trata de aplicaciones orientadas a Web, la lista de opciones parece interminable más aun cuando las empresas buscan paquetes que les permitan manejar sus datos de manera segura, eficiente y sobre todo en cualquier parte del país o del mundo.

Se disponen de herramientas con licenciamiento privado tales como Visual Studio .NET 2005, SQL Server, que brindan una cantidad infinita de soluciones a los problemas que se puedan presentar durante el desarrollo de aplicaciones, además de contar con el respaldo de marcas mundialmente reconocidas como Microsoft.

Visual Studio .NET 2005 es una poderosa herramienta para desarrollo de software en múltiples plataformas y con la posibilidad de trabajar con distintos lenguajes de programación, tales como Basic, c++, ASP entre otros, además de brindar las facilidades de trabajo de un entorno Visual; esta versión de Visual Studio se caracteriza por la integración de muchos tipos de tecnología, para permitir la creación de aplicaciones mucho más robustas y tecnológicamente más avanzadas.

SQL Server es un potente manejador de bases de datos que gracias a su eficiencia le han permitido consolidarse como uno de los mejores en el mercado y ha llevado a muchas de empresas a confiar en él para el manejo de sus datos, además es una aplicación que se integra fácilmente con las distintas herramientas de programación tales como Visual Studio .NET.

1.2.4 Cronograma de actividades.

Tabla 1. Cronograma de actividades

ACTIVIDADES	TIEMPO (SEMANAS)
PLANEACIÓN	5
Inducción a la organización.	1
Revisión de documentos y procesos internos.	1
Capacitación en sistema de gestión de calidad.	1
Revisión y ajuste de plan de trabajo.	1
Estudio de metodologías de desarrollo.	3
Elaboración y entrega de informe de avance	1
EJECUCIÓN	17
Estudio de las nuevas metodologías de diseño y desarrollo.	2
Comparación de las metodologías estudiadas con las utilizadas actualmente.	1
Plantear las adaptaciones a la metodología de diseño y desarrollo actuales.	2
Análisis del impacto de la metodología de DOO con UML en los procesos y procedimientos de la organización.	2
Elaboración y entrega del primer informe de avance etapa de ejecución	1
Apoyo a la construcción de los módulos de facturación y comisiones utilizando Visual Studio .NET	17
Elaboración y entrega del segundo informe de avance de la etapa de ejecución	1

Apoyo a la planeación, ejecución y documentación de pruebas de los módulos de comisiones y facturación.	17
Apoyo a la elaboración de la documentación técnica y de usuarios de los módulos de facturación y comisiones.	17
Elaboración y entrega del tercer informe de avance de la etapa de ejecución	1
Brindar capacitación y soporte técnico a los usuarios de la aplicación desarrollada en su fase de estabilización	8
TERMINACIÓN	2
Transferencia de responsabilidades a cargo.	2
Elaboración y entrega de informe final	1

2. MARCO TEORICO

Se pretende enmarcar dentro de este capítulo las herramientas teóricas utilizadas para el desarrollo de la práctica, haciendo descripciones detalladas del funcionamiento y la estructura de la metodología utilizada, la arquitectura y las nuevas tecnologías .Net que se van a trabajar

2.1 FUNDAMENTACION METODOLOGICA

2.1.1 Metodologías de desarrollo de software. Busca mostrar las principales ventajas y desventajas de algunas de las principales metodologías de desarrollo utilizadas actualmente, así como su funcionamiento

- **Modelo en Cascada.** El modelo en cascada fue uno de los primeros publicados y ha sido la base para otros modelos y procesos de la ingeniería del software actual; consiste en tomar actividades fundamentales del proceso de análisis de requisitos y requerimientos, diseño, desarrollo o construcción, implantación, pruebas y mantenimiento; representándolas como fases separadas del proceso.

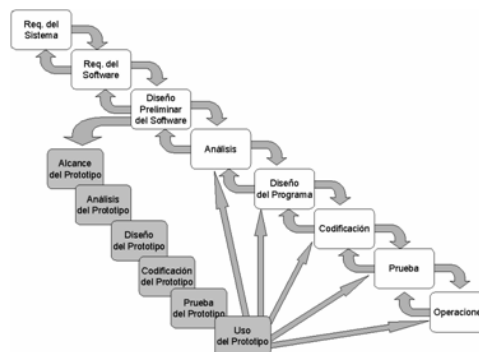
El modelo cascada contempla las siguientes fases

- Fase de análisis de requerimientos y especificación de requisitos; busca definir detalladamente los requerimientos de software y hardware del sistema, así como los servicios, restricciones, y objetivos, establecidos con los usuarios del sistema; en esta fase se realiza un diseño o bosquejo preliminar de la aplicación,

y se busca detectar todas las posibles causas de error y se planifica su corrección.

- Fase de diseño: en esta fase se establece toda la arquitectura del sistema, se identifican y describen abstracciones de todos los procesos y procedimientos, al igual que las relaciones entre los diversos componentes del mismo.
- Fase de desarrollo o construcción: en esta fase se lleva a cabo la construcción y codificación de los módulos y unidades de software, además se realizan las pruebas unitarias a cada uno de los módulos por separado.
- Pruebas e implantación: se realiza el acoplamiento de los distintos componentes del sistema, se realizan pruebas de funcionamiento en conjunto y se le entrega al cliente la solución ya probada y lista para su funcionamiento.
- Operación y mantenimiento: el sistema es puesto en marcha con datos reales y funcionamiento real, se realiza la corrección de errores descubiertos y mejoras de último minuto, además se identifican nuevos requisitos.

Figura 2. Diagrama de modelo en cascada



En la figura 2.1.1.1 se puede observar la interacción existente entre cada una de las fases del modelo, hay que notar que una fase no puede iniciar hasta que no haya sido completamente terminada la anterior, y por lo general se incluye la corrección de problemas encontrados en fases anteriores, y en cada fase se generan documentos que deben ser aprobados por el usuario.

En la practica este modelo no es lineal se requieren varias iteraciones y se regresa varias veces a las fases previas, las principales ventajas de este modelo son

- Una fase no comienza hasta que no hayan terminado las anteriores, esto supone que cada fase ha sido totalmente terminada y que la aparición de errores debe ser mínima
- Esta metodología fue base para las subsecuentes y fue muy aplicada por varias compañías durante mas de dos décadas
- Ayuda a prevenir el sobrepaso en las fechas de entrega y controla sobre costos

Los principales problemas que se presentan con esta metodología son:

- Las iteraciones son costosas en tiempo y cantidad de trabajo, pues implican rehacer debido a la producción y aprobación de documentos.
- Debido al largo tiempo de espera para la entrega del proyecto es altamente probable que el software ya no cumpla con los requerimientos del usuario final.
- Como los problemas que se presentan son relegados para su posterior resolución, es probable que estos sean olvidados o se solucionen de una manera poco ortodoxa

- El cliente pocas veces tiene claros todos los requisitos o se confunde y solicita cosas que en realidad no necesita
- **Prototipado evolutivo.** A continuación se realiza una descripción detallada de esta metodología debido a que fue la metodología escogida para la realización de los objetivos propuestos en esta práctica, más adelante en el numeral 2.1.1 se explicaran los motivos por los cuales se eligió así como sus principales ventajas y desventajas.

Cuando un cliente siente la necesidad de requerir el desarrollo de un Sistema para resolver un problema, habitualmente no posee una idea demasiado detallada de esta necesidad, sólo percibe que tiene un problema que demanda una solución.

Por otra parte, también suele suceder que el ingeniero de software que debe atender a ese cliente, no siempre estará seguro de la viabilidad de la solución que tiene en mente para resolver el problema.

Es en esta situación donde el paradigma de Prototipado Evolutivo adquiere gran valor, ya que permite a ambos realizar una aproximación gradual a la solución óptima del problema.

Existen tres modelos derivados del uso de prototipos:

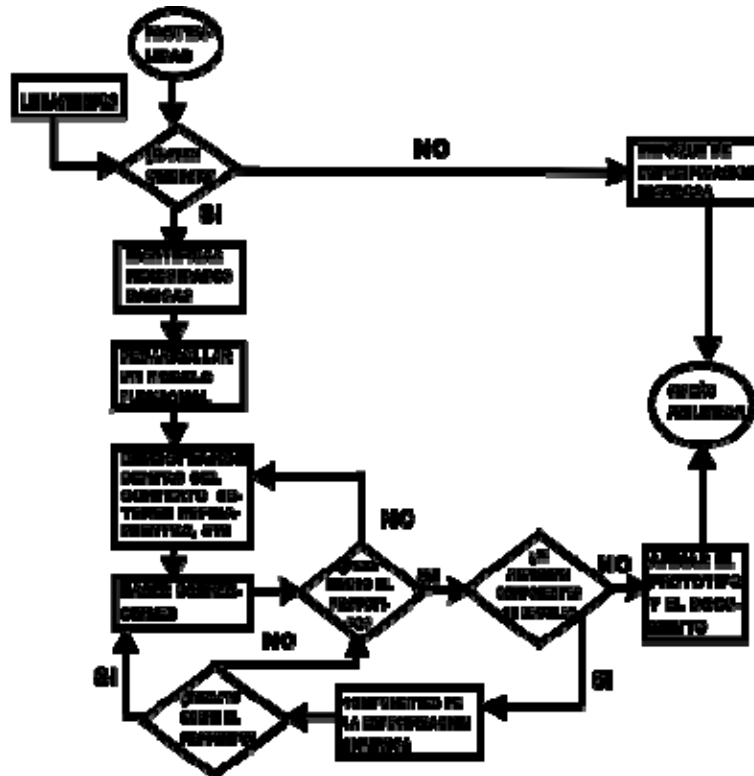
- **Maqueta.** Es un prototipo en papel o modelo que opera sobre una computadora personal que describe exclusivamente la interacción hombre máquina, aportando al usuario un ejemplo visual de las entradas y salidas que tendrá el futuro sistema. Para su confección se utiliza un conjunto de datos de entrada y salida de naturaleza estática.
- **Prototipo desechable.** Esta clase de prototipo también es utilizada para ayudar al cliente a identificar los requisitos del futuro Sistema. En este tipo de

modelos deben implementarse solamente aquellos aspectos del Sistema que se entienden mal o no son del todo conocidos. El usuario, mediante la operación del prototipo, podrá ir refinando y descubriendo aquellos aspectos o requisitos no madurados o desconocidos. Para la operación de este modelo se utilizarán datos reales. Luego de su utilización, todos los elementos de estos prototipos deberán ser descartados.

- **Prototipo evolutivo.** En esta clase de prototipos se desarrolla un modelo de trabajo del Sistema propuesto que debe ser fácilmente modificable y ampliable, permitiendo a los usuarios contar con una representación física inmediata de las partes claves del Sistema antes de arribar al producto definitivo. Una vez definidos todos los requisitos y comprobada la viabilidad de la solución propuesta, el prototipo evolucionará hacia el Sistema final. En este modelo se deben implementar en forma gradual aquellos requisitos y necesidades que vayan resultando claramente interpretados.

De ellos, el modelo más difundido es el de Prototipos Evolutivos que se puede visualizar en la Figura 2.1.1.2.1, ya que implica economía de esfuerzos y la posibilidad que el cliente cuente rápidamente con un modelo básico que le permita solucionar, aunque más no sea, parte del problema. Por tal motivo, es el que utilizaremos en el desarrollo del presente proyecto.

Figura 3. Modelo de Prototipado Evolutivo



Este paradigma reconoce las siguientes fases que conforman el Ciclo de Vida:

- Análisis preliminar y especificación de requisitos.
- Diseño rápido del prototipo.
- Construcción e Implantación - Pruebas.
- Evaluación y refinamiento interactivo del prototipo.
- Refinamiento de las especificaciones del prototipo.
- Producto de Ingeniería e Implantación del Sistema final.

- **Análisis preliminar y especificación de requisitos.** Al igual que en el modelo Clásico, en el de Prototipado Evolutivo el Ciclo de Vida comienza con la especificación de requisitos.

El cliente, junto con el Ingeniero de Software, identifica las necesidades, formulan las soluciones potenciales y evalúan la viabilidad de cada una de las soluciones para seleccionar aquella que parezca más eficiente a nivel de Sistema. Una vez establecidos los límites de la futura aplicación y considerando sólo aquellos Requisitos Globales del Sistema que se han individualizado perfectamente, el Ingeniero de Software analizan con detalle y, junto con el cliente, refinan aquellos requisitos que serán tratados mediante el software. También dejan claramente establecidas las funciones del hardware, del software y de las interfaces, permitiendo, de esa manera, desarrollar la arquitectura básica del prototipo a implementar.

En esta etapa se producen los siguientes elementos:

- ✓ Modelo de la situación actual.
- ✓ Informe preliminar de necesidades (parcial).
- ✓ Estudio de Viabilidad.
- ✓ Especificación de requisitos funcionales del Sistema (HW y SW) (parcial).
- ✓ Descripción del flujo de información (parcial).
- ✓ Especificación de requisitos del software (parcial).
- ✓ Plan de Pruebas (parcial).
- ✓ Especificaciones de las interfaces del Sistema (parcial).
- ✓ Especificaciones de las interfaces del software con otros componentes (parcial).
- ✓ Descripción funcional del Sistema (parcial).
- ✓ Arquitectura del Sistema (parcial).

- **Diseño rápido del prototipo.** Con la información recogida en la etapa anterior, el Ingeniero de Software desarrolla una representación coherente y organizada de aquellos aspectos del sistema software que cumplen con las especificaciones de los requisitos que se han podido recoger y refinar en total acuerdo con el cliente.

Dado que esta fase requiere la obtención de un diseño rápido para ser posteriormente implementado, las etapas de diseño de alto nivel y diseño detallado vistas en el ciclo de vida tradicional, se funden en una sola etapa de diseño.

En ella, se focaliza con la misma profundidad en las funciones y estructuras de los componentes que conforman el sistema, como en la definición detallada de los flujos datos y control. También deben refinarse las representaciones algorítmicas que se utilizan en cada componente modular y las interfaces que requiere el prototipo a desarrollar, simulando, de resultar necesario, la presencia de otros módulos o dispositivos que a esta altura del proyecto aún no han sido desarrolladas o se duda de su necesidad de implementación.

Por tratarse de un prototipo evolutivo, debe prestarse especial atención a la interfase hombre – maquina, ya que, en la medida que el usuario se sienta cómodo en la operación del prototipo, más disposición y tiempo tendrá para ir madurando sus reales necesidades y descubriendo nuevos requisitos que no se manifestaron en la definición inicial.

Los elementos a obtener en esta fase son:

- ✓ Descripción de diseño del Software (parcial).
- ✓ Descripción de la arquitectura. (parcial).
- ✓ Descripción de las bases de datos - si es de aplicación - (parcial).

- ✓ Descripción de interfaces (parcial).
- ✓ Descripción de algoritmos (parcial).
- ✓ Referencias cruzadas con los requisitos.
- ✓ Previsiones de pruebas.

- **Construcción e implantación - Pruebas.** En el modelo de Construcción de Prototipos, en esta fase se unifican las fases de Construcción y Pruebas del modelo Clásico.

En lo que respecta a construcción, al igual que en el modelo Clásico, en esta fase se traduce el diseño a un lenguaje de programación que pueda ser luego interpretado por la computadora.

El código fuente debe estar acompañado de la documentación correspondiente de acuerdo a los estándares y metodologías adoptados para el proyecto.

Para el caso que el prototipo esté conformado por componentes Hardware y Software, en esta fase también se debe planificar y ejecutar la integración de ambos componentes.

Esta fase contempla también la etapa de pruebas, que constituye la revisión final de las especificaciones, el diseño y la codificación.

Para las pruebas, se ejecutará el prototipo con datos de entrada prefijados para comprobar si los resultados que el Sistema desarrollado produce son similares a los que, teóricamente y según las especificaciones, el Sistema debería producir detectando así posibles errores.

A la finalización de esta fase se obtienen los siguientes elementos:

- ✓ Listados y archivo sobre soporte magnético del código fuente.
- ✓ Listados y archivo sobre soporte magnético de las bases de datos generadas.
- ✓ Archivo objeto generado.
- ✓ Documentación del Prototipo.
- ✓ Documentación del usuario (parcial).
- ✓ Plan de Integración (parcial).
- ✓ Sistema software integrado (parcialmente).
- ✓ Documentación relacionada con las herramientas de HW y SW (características de las máquinas empleadas para el desarrollo, compiladores, linkeadores, depuradores) utilizados para generar el producto.
- ✓ Datos para la fase de pruebas.
- ✓ Especificación de las pruebas del prototipo.
- ✓ Informe resumen de pruebas del prototipo.

- **Evaluación y refinamiento interactivo del prototipo.** Es en esta fase donde el cliente / usuario toma contacto con el prototipo para, mediante su utilización, evaluar su utilidad con datos reales, aunque sólo sea en forma parcial, para resolver el problema.

Se produce aquí un proceso interactivo entre el cliente / usuario y el desarrollador, en el cuál, a partir de la experiencia en la utilización del prototipo, se depuren conceptos, surjan nuevas ideas o se detecten fallos de interpretación en requisitos o alcances de determinados servicios que debe prestar el futuro sistema.

A la finalización de esta fase pueden suceder dos cosas:

- Que aún no se haya llegado a satisfacer los requisitos del cliente y se disponga de recursos para continuar con el proyecto, procediéndose, entonces, a la generación de un nuevo prototipo, creando los siguientes elementos y pasando a la fase e):

Correcciones a aplicar en:

- ✓ Informe preliminar de necesidades.
- ✓ Especificación de requisitos funcionales del Sistema (HW y SW).
- ✓ Flujo de Información.
- ✓ Especificación de requisitos del software.
- ✓ Especificaciones de las interfaces del Sistema.
- ✓ Especificación de nuevos requisitos funcionales del Sistema.

- Que ya se haya hecho más de una iteración y se haya comprobado que el prototipo desarrollado satisface los requisitos del cliente (o se han agotado los recursos destinados al proyecto), se pasará directamente a la fase VI).

• **Refinamiento de las especificaciones del prototipo.** Con los elementos generados en la fase anterior, se realiza un refinamiento de las especificaciones que servirán como base de una nueva fase de diseño rápido, con el objeto de generar un nuevo prototipo que contemplará mayores prestaciones que el anterior.

Los elementos a generados en esta fase son los siguientes:

- ✓ Informe preliminar de necesidades (parcial).
- ✓ Especificación de requisitos funcionales del Sistema (HW y SW) (parcial).
- ✓ Descripción del flujo de información (parcial)

- ✓ Especificación de requisitos del software (parcial)
- ✓ Especificaciones de las interfaces del Sistema (parcial).
- ✓ Especificaciones de las interfaces del software con otros componentes (parcial).
- ✓ Descripción funcional del Sistema (parcial).
- ✓ Arquitectura del Sistema (parcial).

Con estos elementos se vuelve a la fase II Diseño Rápido, para generar un nuevo prototipo con mayores prestaciones.

- **Producto de Ingeniería - Implantación del Sistema final.** Esta fase constituye la etapa final del desarrollo del modelo de prototipado evolutivo.

Comprende las tareas de consolidación técnica del producto, pruebas y revisiones finales, instalación y operación, debiéndose adoptar las provisiones necesarias para actividades de soporte y remoción del Producto.

Esta fase se continúa hasta el momento en que el Sistema es retirado de operación para ser reemplazado por un nuevo Sistema.

Durante esta fase se producen los siguientes elementos:

- ✓ Documentación del Sistema.
- ✓ Informe resumen de pruebas.
- ✓ Documentación del usuario.
- ✓ Plan de Integración.
- ✓ Documentación relacionada con las herramientas de HW y SW (características de las máquinas empleadas para el desarrollo, compiladores, linkeadores, depuradores) utilizados para generar el producto.
- ✓ Sistema software integrado.

- ✓ Plan de Mantenimiento.
- ✓ Histórico de pedidos de mantenimiento.
- ✓ Ordenes de Mantenimiento.
- ✓ Documentación relacionada con la Gestión de Configuración.
- ✓ Recomendaciones de Mantenimiento.
- ✓ Plan de retiro.

Los productos parciales de la fase II y III pasan a ser finales.

Las principales ventajas de esta metodología de desarrollo se enuncian a continuación:

- Permite mejorar la definición de los requisitos; debido a que el cliente no los alcanza a especificar todos al principio, se pueden agregar nuevos requisitos, en la medida que se van generando los nuevos prototipos
- Se van cumpliendo las necesidades inmediatas del cliente y se pueden corregir o modificar procedimientos que ya no sean necesarios
- La calidad del software mejora en la medida que se van cumpliendo con los requisitos solicitados por el usuario inmediatamente
- No genera mucho traumatismo la corrección de un error o el acoplamiento de un cambio

Los problemas que se generan normalmente con este tipo de metodología son:

- Hay que documentar cada versión del sistema, lo cual se vuelve poco efectivo para sistemas en los que se requiere que el desarrollo sea rápido.
- Se pueden incrementar los costos de mantenimiento debido a que los cambios continuos pueden llegar a perjudicar la estructura del software.
- Algunos módulos funcionales pero ineficientes puede pasar a convertirse en parte del sistema.
- Se necesita el uso de técnicas y herramientas especiales que faciliten el desarrollo del prototipado, que pueden llegar a ser incompatibles o poco conocidas
- Este tipo de metodología es recomendada sobretodo para sistemas y proyectos de tamaño mediano y pequeño, que requieran desarrollos rápidos, que no se tenga acceso de lleno a los requisitos y en los que existan dudas acerca de alguna parte del sistema
- **Modelo en Espiral.** Modelo propuesto por Boehm⁴ en el año de 1988; el modelo se representa como una espiral en lugar de cómo un conjunto de actividades sucesivas con retrospectiva entre actividades; es decir que cada fase del proceso es representada por una vuelta del espiral, por lo tanto no hay actividades fijas, cada vuelta del espiral determina sus propias tareas a realizar.

En este modelos se denominan sectores a las divisiones del espiral, revisar grafica 2.1.1.3, cada uno de estos sectores especifica cada fase a desarrollar dentro de la versión o ciclo del espiral, estas fases se agrupan así:

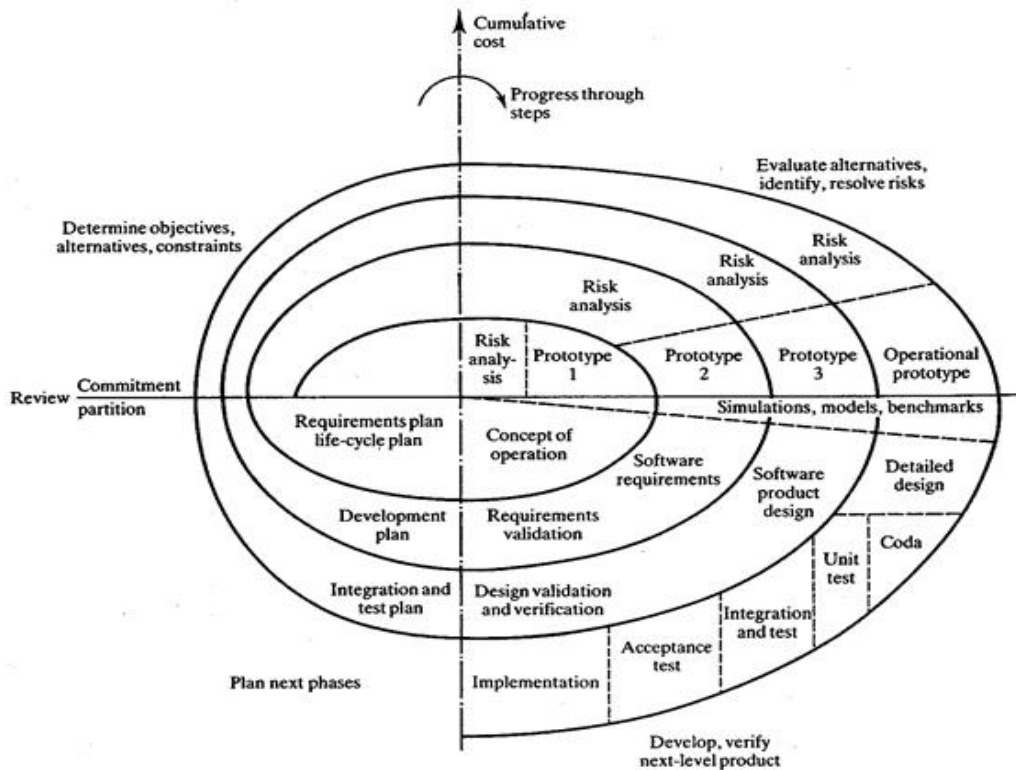
⁴ Boehm, B. W., A Spiral Model of Software Development and Enhancement, IEEE Computer, 1988.

- Definición de objetivos: en este sector se definen los objetivos de cada ciclo, así como la identificación de restricciones y riesgos del producto y dependiendo de estos se elaboran las posibles estrategias y alternativas de desarrollo, además se realiza un diseño detallado del plan administrativo
- Evaluación y reducción de riesgos: se toman cada uno de los riesgos identificados en la fase anterior y se realiza un análisis detallado, generando un conjunto de actividades que permitan minimizarlo
- Desarrollo y validación: después de evaluados los riesgos se escoge el modelo de desarrollo más apropiado de entre los modelos genéricos, esta escogencia depende del riesgo identificado para esta fase
- Planificación: se revisa el proyecto y se decide la continuidad del siguiente ciclo, de ser afirmativo se realiza la planificación de la siguiente etapa del proyecto (o vuelta en el espiral).

Algunas de las principales ventajas de este modelo son las siguientes

- Esta dirigido especialmente a la administración del proyecto
- Tiene en cuenta sobretodo la evaluación y minimización de riesgos que entorpezcan el desarrollo del proyecto lo que evita retrasos

Figura 4. Diagrama del Modelo en espiral



Algunas de las desventajas de este modelo son:

- Se necesita mucha interacción con el cliente y el usuario final lo que implica, en algunos casos, depender de él para realizar cambios o avanzar a un nuevo ciclo, a veces el cliente no tiene la disponibilidad de tiempo ni de recursos para trabajar mancomunadamente con los desarrolladores del proyecto.
- Genera altos costos, pues toda la realización de estudios para minimizar los riesgos necesita una gran cantidad de recursos físicos como humanos

- **DRA Diseño Rápido de Aplicaciones.** Consiste en un proceso lineal secuencial que enfatiza el desarrollo de software extremadamente rápido; esta “alta velocidad” se logra utilizando un enfoque de construcción basado en componentes. Si se conocen al detalle todos los requisitos, pueden lograr entregas de proyectos totalmente funcionales en plazos cortos de tiempo.

Cuando se utiliza principalmente para aplicaciones de sistemas de información, el enfoque DRA comprende las siguientes fases:

- Modelado de gestión: el flujo de información entre las funciones de gestión se modela de forma que responda a las siguientes preguntas: ¿Qué información conduce el proceso de gestión? ¿Qué información se genera? ¿Quién la genera? ¿A dónde va la información? ¿Quién la proceso?
- Modelado de datos: el flujo de información definido como parte de la fase de modelado de gestión se refina como un conjunto de objetos de datos necesarios para apoyar la empresa. Se definen las características (llamadas atributos) de cada uno de los objetos y las relaciones entre estos objetos.
- Modelado de proceso: los objetos de datos definidos en la fase de modelado de datos quedan transformados para lograr el flujo de información necesario para implementar una función de gestión. Las descripciones del proceso se crean para añadir, modificar, suprimir, o recuperar un objeto de datos. Es la comunicación entre los objetos.
- Generación de aplicaciones: El DRA asume la utilización de técnicas de cuarta generación. En lugar de crear software con lenguajes de programación de tercera generación, el proceso DRA trabaja para volver a utilizar componentes de programas ya existentes (cuando es posible) o a crear componentes reutilizables (cuando sea necesario). En todos los casos se utilizan herramientas automáticas para facilitar la construcción del software.
-

- Pruebas de entrega: Como el proceso DRA enfatiza la reutilización, ya se han comprobado muchos de los componentes de los programas. Esto reduce tiempo de pruebas. Sin embargo, se deben probar todos los componentes nuevos y se deben ejercitar todas las interfases a fondo.

Estas son algunas de las ventajas que tiene el modelo DRA:

- Velocidad de entrega de los proyectos, permite cumplir con cronogramas que no cuentan con mucho tiempo para el desarrollo de las aplicaciones.
- Reutiliza componentes que ya han sido funcionales, lo cual evita rehacer trabajos lo cual implica costos y demoras

También como en todos los modelos el DRA presenta algunos inconvenientes:

- Se necesita gran cantidad de recursos que permitan lograr crear los equipos de trabajo necesarios para proyectos grandes.
- No es muy recomendado para proyectos en los cuales los riesgos técnicos son bastante altos.
- DRA requiere clientes y desarrolladores comprometidos en las rápidas actividades necesarias para completar un sistema en un marco de tiempo abreviado.
- **XP eXtreme Programming.** Centrarse principalmente en dimensiones como el factor humano o el producto software es la filosofía de las metodologías ágiles, las cuales dan mayor valor al individuo, a la relación con el cliente y al desarrollo incremental de software con iteraciones muy cortas; programación extrema (eXtreme Programming XP) hace parte de esta filosofía, siendo actualmente la mas popular.

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en

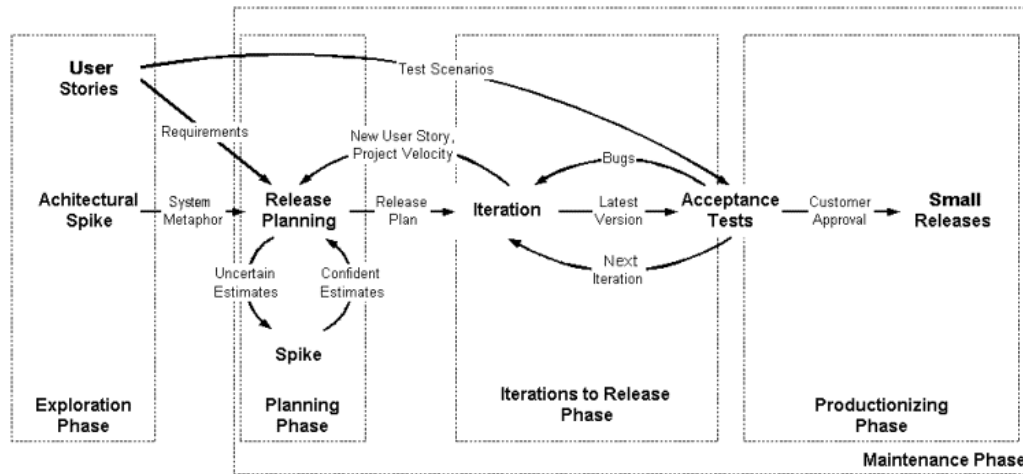
equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Proceso XP. Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos [12]:

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

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, 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.

Figura 5. Diagrama de modelo XP



- **Fase I: Exploración.** En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

- **Fase II: Planificación de la Entrega.** En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.

- **Fase III: Iteraciones.** Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción.
- **Fase IV: Producción.** La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.
- **Fase V: Mantenimiento.** Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.
- **Fase VI: Muerte del Proyecto.** Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

Algunas de las principales ventajas se enumeran a continuación

- Se consigue un producto que funciona pronto con los beneficios que esto conlleva para el cliente, comprende el sistema y puede introducir mejoras que lo hacen eficaz.
- Los equipos ágiles tienen una gran comunicación tanto los desarrolladores como los expertos del negocio (cambios continuos en la tecnología).
- La XP agrupa todas las técnicas y pone todo su énfasis en realizar pruebas, donde cada programador escribe sus pruebas conforme desarrolla software.

Entre los principales problemas se encuentran:

- El problema de los modelos ágiles es que requieren de un equipo eficaz de desarrollo, tanto a nivel individual como de equipo.
- En los proyectos tradicionales el personal puede ser reemplazado, sin embargo en los modelos ágiles esto varía, los desarrolladores pueden tomar todas las decisiones.
- **RUP Rational Unified Process.**
 - **Características esenciales.** Los autores de RUP destacan que el proceso de software propuesto por RUP tiene tres características esenciales: está dirigido por los Casos de Uso, está centrado en la arquitectura, y es iterativo e incremental.

- **Proceso dirigido por Casos de Uso.** Según [Kru00], los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema.

En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba. Los Casos de Uso constituyen un elemento integrador y una guía del trabajo como se muestra en la Figura 2.

- **Proceso centrado en la arquitectura.** La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo [Kru00].

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema.

En el caso de RUP además de utilizar los Casos de Uso para guiar el proceso se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.

Cada producto tiene tanto una función como una forma. La función corresponde a la funcionalidad reflejada en los Casos de Uso y la forma la proporciona la arquitectura. Existe una interacción entre los Casos de Uso y la arquitectura, los Casos de Uso deben encajar en la arquitectura cuando se llevan a cabo y la arquitectura debe permitir el desarrollo de todos los Casos de Uso requeridos, actualmente y en el futuro. Esto provoca que tanto arquitectura como Casos de Uso deban evolucionar en paralelo durante todo el proceso de desarrollo de software.

Es conveniente ver el sistema desde diferentes perspectivas para comprender mejor el diseño por lo que la arquitectura se representa mediante varias vistas que se centran en aspectos concretos del sistema, abstrayéndose de los demás. Para RUP, todas las vistas juntas forman el llamado modelo 4+1 de la arquitectura [Kru95], el cual recibe este nombre porque lo forman las vistas lógica, de implementación, de proceso y de despliegue, más la de Casos de Uso que es la que da cohesión a todas.

Al final de la fase de elaboración se obtiene una *baseline*⁵ de la arquitectura donde fueron seleccionados una serie de Casos de Uso arquitectónicamente relevantes (aquellos que ayudan a mitigar los riesgos más importantes, aquellos que son los más importantes para el usuario y aquellos que cubran las funcionalidades significativas)

⁵ Una baseline es una instantánea del estado de todos los artefactos del proyecto, registrada para efectos de gestión de configuración y control de cambios.

Como se observa en la Figura 5, durante la construcción los diversos modelos van desarrollándose hasta completarse (según se muestra con las formas rellenas en la esquina superior derecha). La descripción de la arquitectura sin embargo, no debería cambiar significativamente (abajo a la derecha) debido a que la mayor parte de la arquitectura se decidió durante la elaboración. Se incorporan pocos cambios a la arquitectura (indicados con mayor densidad de puntos en la figura inferior derecha) [JBR00]

- **Proceso iterativo e incremental.** Según [JBR00] el equilibrio correcto entre los Casos de Uso y la arquitectura es algo muy parecido al equilibrio de la forma y la función en el desarrollo del producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.

Una iteración puede realizarse por medio de una cascada como se muestra en la Figura 6. Se pasa por los flujos fundamentales (Requisitos, Análisis, Diseño, Implementación y Pruebas), también existe una planificación de la iteración, un análisis de la iteración y algunas actividades específicas de la iteración. Al finalizar se realiza una integración de los resultados con lo obtenido de las iteraciones anteriores.

El proceso iterativo e incremental consta de una secuencia de iteraciones. Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando

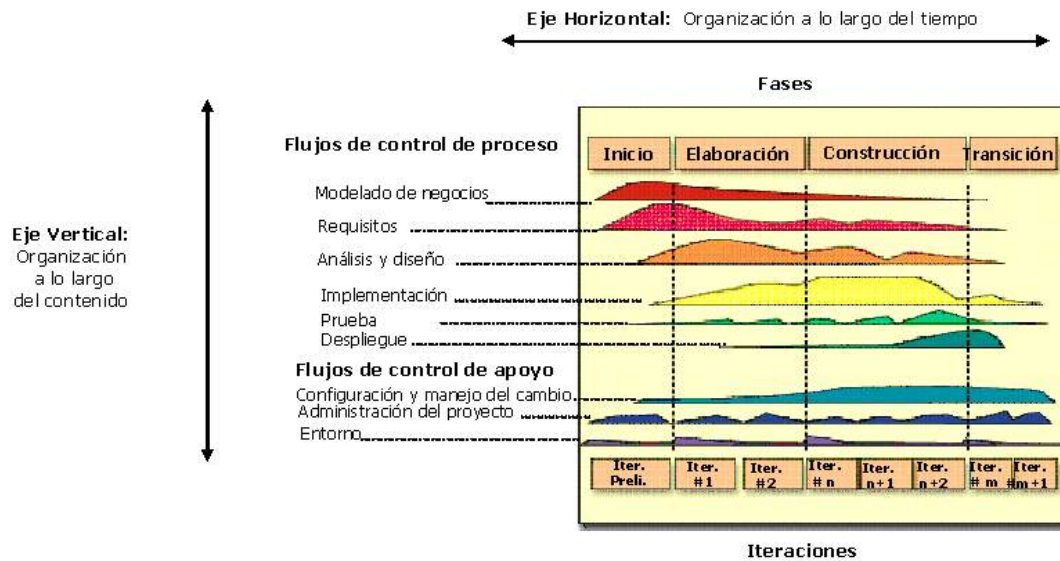
termina. Se puede determinar si han aparecido nuevos requisitos o han cambiado los existentes, afectando a las iteraciones siguientes. Durante la planificación de los detalles de la siguiente iteración, el equipo también examina cómo afectarán los riesgos que aún quedan al trabajo en curso. Toda la retroalimentación de la iteración pasada permite reajustar los objetivos para las siguientes iteraciones. Se continúa con esta dinámica hasta que se haya finalizado por completo con la versión actual del producto.

- **Estructura del proceso**

El proceso puede ser descrito en dos dimensiones o ejes [RSC98]:

- **Eje horizontal.** Representa el tiempo y es considerado el eje de los aspectos dinámicos del proceso. Indica las características del ciclo de vida del proceso expresado en términos de fases, iteraciones e hitos. Se puede observar en la Figura 8 que RUP consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Como se mencionó anteriormente cada fase se subdivide a la vez en iteraciones.
- **Eje vertical.** Representa los aspectos estáticos del proceso. Describe el proceso en términos de componentes de proceso, disciplinas, flujos de trabajo, actividades, artefactos y roles.

Figura 6. Estructura de RUP



- Estructura Dinámica del proceso. Fases e iteraciones.** RUP se repite a lo largo de una serie de ciclos que constituyen la vida de un producto. Cada ciclo concluye con una generación del producto para los clientes. Cada ciclo consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada fase se subdivide a la vez en iteraciones, el número de iteraciones en cada fase es variable.

Cada fase se concluye con un hito bien definido, un punto en el tiempo en el cual se deben tomar ciertas decisiones críticas y alcanzar las metas clave antes de pasar a la siguiente fase, ese hito principal de cada fase se compone de hitos menores que podrían ser los criterios aplicables a cada iteración. Los hitos para cada una de las fases son: Inicio - *Lifecycle Objectives*, Elaboración - *Lifecycle Architecture*, Construcción - *Initial Operational Capability*, Transición - *Product Release*.

- **Inicio.** Durante la fase de inicio se define el modelo del negocio y el alcance del proyecto. Se identifican todos los actores y Casos de Uso, y se diseñan los Casos de Uso más esenciales (aproximadamente el 20% del modelo completo). Se desarrolla, un plan de negocio para determinar que recursos deben ser asignados al proyecto.

- **Elaboración.** El propósito de la fase de elaboración es analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del proyecto y eliminar los mayores riesgos.

En esta fase se construye un prototipo de la arquitectura, que debe evolucionar en iteraciones sucesivas hasta convertirse en el sistema final. Este prototipo debe contener los Casos de Uso críticos identificados en la fase de inicio. También debe demostrarse que se han evitado los riesgos más graves.

En esta fase se debe tratar de abarcar todo el proyecto con la profundidad mínima. Sólo se profundiza en los puntos críticos de la arquitectura o riesgos importantes.

En la fase de elaboración se actualizan todos los productos de la fase de inicio.

- **Construcción.** La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones. Durante esta fase todos los componentes, características y requisitos deben ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del producto.

- **Transición.** La finalidad de la fase de transición es poner el producto en manos de los usuarios finales, para lo que se requiere desarrollar nuevas versiones actualizadas del producto, completar la documentación, entrenar al usuario en el manejo del producto, y en general tareas relacionadas con el ajuste, configuración, instalación y facilidad de uso del producto.
- **Otras prácticas.** RUP identifica 6 *best practices* con las que define una forma efectiva de trabajar para los equipos de desarrollo de software.
- **Gestión de requisitos.** RUP brinda una guía para encontrar, organizar, documentar, y seguir los cambios de los requisitos funcionales y restricciones. Utiliza una notación de Caso de Uso y escenarios para representar los requisitos.
- **Desarrollo de software iterativo.** Desarrollo del producto mediante iteraciones con hitos bien definidos, en las cuales se repiten las actividades pero con distinto énfasis, según la fase del proyecto.
- **Desarrollo basado en componentes.** La creación de sistemas intensivos en software requiere dividir el sistema en componentes con interfaces bien definidas, que posteriormente serán ensamblados para generar el sistema. Esta característica en un proceso de desarrollo permite que el sistema se vaya creando a medida que se obtienen o se desarrollan sus componentes.
- **Modelado visual (usando UML)**

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema software. Es un estándar de la OMG. Utilizar herramientas de modelado visual facilita la gestión de dichos modelos, permitiendo ocultar o exponer detalles cuando sea necesario. El modelado visual también

ayuda a mantener la consistencia entre los artefactos del sistema: requisitos, diseños e implementaciones. En resumen, el modelado visual ayuda a mejorar la capacidad del equipo para gestionar la complejidad del software.

- **Verificación continua de la calidad.** Es importante que la calidad de todos los artefactos se evalúe en varios puntos durante el proceso de desarrollo, especialmente al final de cada iteración. En esta verificación las pruebas juegan un papel fundamental y se integran a lo largo de todo el proceso. Para todos los artefactos no ejecutables las revisiones e inspecciones también deben ser continuas.

- **Gestión de los cambios.** El cambio es un factor de riesgo crítico en los proyectos de software. Los artefactos software cambian no sólo debido a acciones de mantenimiento posteriores a la entrega del producto, sino que durante el proceso de desarrollo, especialmente importantes por su posible impacto son los cambios en los requisitos. Por otra parte, otro gran desafío que debe abordarse es la construcción de software con la participación de múltiples desarrolladores, posiblemente distribuidos geográficamente, trabajando a la vez en una *release*, y quizás en distintas plataformas. La ausencia de disciplina rápidamente conduciría al caos. La Gestión de Cambios y de Configuración es la disciplina de RUP encargada de este aspecto.

2.1.2 Metodologías Utilizadas. La empresa ha venido trabajando con un modelo en cascada, en aplicaciones Windows desarrolladas en Visual Basic 6.0, en las que se trabajó con objetos y se utilizó metodología de clases; se desea llegar a una metodología más refinada, para aprovechar al máximo las ventajas que ofrece la herramienta para el manejo de objetos, diagramas de clases, herencias,

asociaciones, que facilitan la siguiente etapa, la implementación, generando así un ahorro considerable de tiempo y esfuerzo por parte de los desarrolladores.

Evaluando los impactos técnicos y económicos que implica el cambio de una metodología de desarrollo y teniendo en cuenta que la aplicación principal que se viene desarrollando ya tiene algunos procesos avanzados, se define usar una adaptación de las metodologías RUP y Prototipado Evolutivo, con la metodología actual existente en la empresa.

En términos generales se dan a la empresa algunas recomendaciones sobre la actualización de su metodología de desarrollo de software para que esta se adapte mejor con las nuevas tecnologías en las que la empresa esta incursionando y en la medida de las capacidades y avances se realizaran las respectivas modificaciones evitando un cambio brusco que vaya en detrimento de los ingresos de la institución.

- **Plantear las adaptaciones a la metodología de diseño y desarrollo actuales.** Se realizaron algunas pruebas piloto, en otras aplicaciones que esta desarrollando la empresa actualmente para probar las ventajas y desventajas que ofrecen las nuevas metodologías de diseño y así plantear los cambios que deben plantearse a las existentes.

Con estas pruebas se realizaron algunos prototipos de aplicaciones más pequeñas orientadas Web solicitadas por otros clientes; con los resultados obtenidos se llego a la conclusión que los cambios mas importantes deben realizarse en la fase de diseño.

- **Análisis del impacto de la metodología a los procesos y procedimientos de la organización.** En el desarrollo de estas pruebas se encontró que el uso de UML facilita y reduce el tiempo que se debe dedicar a las labores de diseño y que evitan permiten un diseño bastante mas completo lo que evita la aparición de errores o problemas en el diseño de la aplicación cuando ya se este llevando a cabo la fase de desarrollo, problema bastante frecuente y que genera muchos inconvenientes en el cumplimiento de los cronogramas de trabajo del proyecto.

2.2 APLICACIONES DISTRIBUIDAS

Las aplicaciones han pasado por un proceso evolutivo enorme. Desde su inicio con las aplicaciones monolíticas donde en una aplicación todo estaba ligado o mezclado por decirlo de alguna manera. Es decir, la interfaces de usuario, la lógica de cómo funcionaba la empresa y el manejo de la información almacenada y recuperada estaban juntas.

Luego la industria ha implementado un nuevo modelo de aplicaciones, las aplicaciones distribuidas cliente/servidor, que se convirtió en el estándar por un tiempo. Pero con la llegada de las aplicaciones Web se hacia necesario un nuevo estándar para la operaciones de los sistemas, y es por esto que se ha propuesto el modelo de las aplicaciones en n-capas.

Este modelo por lo general esta basado en un esquema de tres partes: Acceso, Lógica de negocios e interfaces de usuario. Aunque es posible continuar subdividiendo este modelo en sub-capas para una mayor flexibilidad en la distribución en el equipo de desarrollo y durante el mantenimiento. Veamos en que consiste.

En nuestros días el vertiginoso avance de la informática y las comunicaciones, con sus mayores exponentes Internet e Intranet, y la cada vez más creciente demanda de la empresa de aplicaciones de calidad que den solución a sus necesidades, ha hecho que las técnicas tradicionales de diseño e implementación de aplicaciones comiencen hacer insuficiente, por lo que un nuevo enfoque de desarrollo se hace necesario.

Los arquitectos de aplicaciones pueden usar la plataforma .NET para desarrollar, implementar y dar asistencia a las aplicaciones distribuidas. Esta plataforma, altamente integrada, pero flexible, permite a los programadores generar soluciones empresariales completas que permiten aprovechar las arquitecturas y aplicaciones existentes.

Windows DNA era una arquitectura para generar aplicaciones distribuidas basadas en Web y de estricta correspondencia. Cuando las aplicaciones distribuidas empezaron a requerir unos principios de correspondencia menos precisos, la arquitectura de Microsoft cambió a la de la plataforma .NET.

2.2.1 Filosofía y ventajas. El principio fundamental de las aplicaciones distribuidas es la división lógica de una aplicación en tres capas fundamentales:

Presentación

Lógica empresarial

Acceso a datos y almacenamiento

Los programadores pueden generar aplicaciones altamente escalables y flexibles, organizando las aplicaciones según estos criterios. Para ello, deberán usar técnicas de programación basadas en componentes y emplear al máximo las características de la plataforma .NET y del sistema operativo Microsoft Windows.

Un modelo simple de aplicación distribuida consiste en un cliente que comunica con la capa intermedia, que a su vez es el servidor de aplicaciones y una aplicación que contiene la lógica empresarial. La aplicación, a su vez, comunica con una base de datos que suministra y almacena datos.

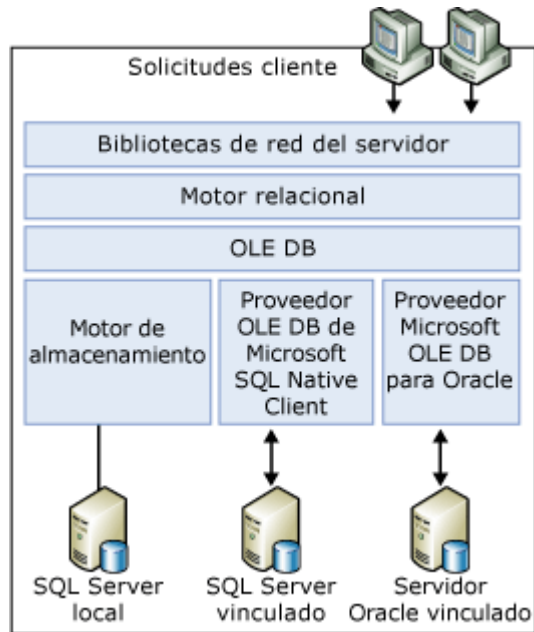
- **Servicios de presentación.** La capa de presentación incluye tanto una interfaz de cliente enriquecida (o sencilla) como una aplicación. El cliente enriquecido proporciona al sistema operativo — ya sea directamente mediante la API Win32 de Microsoft o indirectamente mediante formularios Windows Forms— una interfaz completa de programación, y usa ampliamente sus componentes. El cliente sencillo (explorador Web) se está convirtiendo rápidamente en la interfaz preferida de muchos programadores. Un programador puede generar lógica empresarial que pueda ejecutarse en cualquiera de los tres niveles de la aplicación. Con las aplicaciones para Web ASP.NET y los servicios Web XML, el cliente sencillo puede proporcionar a las aplicaciones una interfaz de usuario visualmente rica, flexible e interactiva. Los clientes sencillos también tienen la ventaja de aportar un mayor grado de portabilidad entre plataformas.

- **Lógica empresarial/ Servicios de la aplicación.** Esta capa se divide en servidores de aplicaciones y servicios, disponibles para ayudar a los clientes. Las aplicaciones Web pueden escribirse para sacar partido de los servicios COM+, de Message Queuing (MSMQ), de los servicios de directorio y de los servicios de seguridad mediante .NET Framework. Los servicios de la aplicación, por el contrario, pueden interactuar con multitud de servicios en la capa de acceso a datos.

- **Acceso a datos y servicios de almacenamiento.** Los servicios de datos que admiten acceso a datos y almacenamiento son:
 - ADO.NET, que proporciona un acceso simplificado a los datos mediante programación, ya sea por medio de lenguajes de secuencias de comandos o de programación.
 - OLE DB, que es un proveedor de datos universal desarrollado por Microsoft.
 - XML, que es un formato estándar para especificar estructuras de datos. XML es un estándar escogido por la comunidad internauta. Mientras que HTML se centra en el modo en el que el explorador procesa la información y la presenta en pantalla, el objetivo de XML es manejar la estructura de datos y su representación.

- **Servicios del sistema.** .NET Framework y el sistema operativo Windows aceptan en su totalidad los elementos que conforman cada segmento de este modelo. Entre sus muchos servicios se encuentran los de directorio, seguridad, administración y comunicaciones, que operan en las tres capas. Las herramientas de programación, que incluyen el sistema de desarrollo Visual Studio .NET, permiten a los programadores generar componentes de aplicaciones entre las diferentes capas.

Figura 7. Diagrama de Aplicaciones distribuidas



2.3 APLICACIONES WEB

En sus inicios las páginas Web eran simplemente documentos estáticos que mostraban al usuario una cantidad de información sobre un tema determinado, pero que no permitían la interacción del usuario con esta información o con información nueva que él mismo necesitara, actualmente con la necesidad imperiosa de tener la información disponible desde cualquier lugar del mundo y poder interactuar con ella ha generado una gran demanda de tener aplicaciones robustas, rápidas fuertes y que permitan un manejo fiable de la información desde cualquier lugar del mundo.

Las aplicaciones Web se han convertido en pocos años en complejos sistemas con interfaces de usuario cada vez más parecidas a las aplicaciones de escritorio,

dando servicio a procesos de negocio de considerable envergadura y estableciéndose sobre ellas requisitos estrictos de accesibilidad y respuesta. Esto ha exigido reflexiones sobre la mejor arquitectura y las técnicas de diseño más adecuadas.

2.3.1 funcionamiento. En los últimos años, la rápida expansión de Internet y del uso de intranets corporativas ha supuesto una transformación en las necesidades de información de las organizaciones. En particular esto afecta a la necesidad de que:

- La información sea accesible desde cualquier lugar dentro de la organización e incluso desde el exterior.
- Esta información sea compartida entre todas las partes interesadas, de manera que todas tengan acceso a la información completa (o a aquella parte que les corresponda según su función) en cada momento.

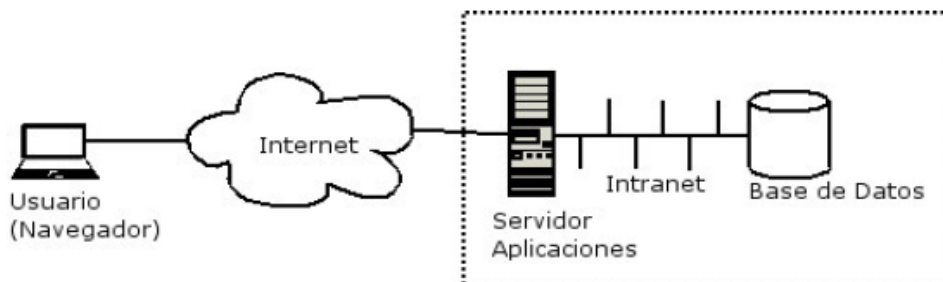
Estas necesidades han provocado un movimiento creciente de cambio de las aplicaciones tradicionales de escritorio hacia las aplicaciones Web, que por su idiosincrasia, cumplen a la perfección con las necesidades mencionadas anteriormente. Por tanto, los sitios Web tradicionales que se limitaban a mostrar información se han convertido en aplicaciones capaces de una interacción más o menos sofisticada con el usuario. Inevitablemente, esto ha provocado un aumento progresivo de la complejidad de estos sistemas y, por ende, la necesidad de buscar opciones de diseño nuevas que permitan dar con la arquitectura óptima que facilite la construcción de los mismos.

Una ventaja significativa en la construcción de aplicaciones Web que soporten las características de los browsers estándar es que deberían funcionar igual

independientemente de la versión del sistema operativo instalado en el cliente. En vez de crear clientes para Windows, Mac OS X, GNU/Linux, y otros sistemas operativos, la aplicación es escrita una vez y es mostrada casi en todos lados. Sin embargo, aplicaciones inconsistentes de HTML, CSS, DOM y otras especificaciones de browsers pueden causar problemas en el desarrollo y soporte de aplicaciones Web. Adicionalmente, la habilidad de los usuarios a personalizar muchas de las características de la interfaz (como tamaño y color de fuentes, tipos de fuentes, inhabilitar Javascript) puede interferir con la consistencia de la aplicación Web.

Otra aproximación es utilizar *Macromedia Flash* o *Java applets* para producir parte o toda la interfaz de usuario. Como casi todos los browsers incluyen soporte para estas tecnologías (usualmente por medio de plug-ins), aplicaciones basadas en Flash o Java pueden ser implementadas con aproximadamente la misma facilidad. Como hacen caso omiso de las configuraciones de los browsers estas tecnologías permiten más control sobre la interfaz, aunque incompatibilidad entre implementaciones de Flash o Java puedan traer nuevas complicaciones. Por las similitudes con una arquitectura cliente-servidor, con un cliente un poco “especializado”, hay disputas sobre si llamar a estos sistemas “aplicaciones Web”; un término alternativo es “aplicación enriquecida de Internet”.

Figura 8. Funcionamiento de Aplicaciones Web



2.3.2 Arquitectura. El usuario interactúa con las aplicaciones Web a través del navegador. Como consecuencia de la actividad del usuario, se envían peticiones al servidor, donde se aloja la aplicación y que normalmente hace uso de una base de datos que almacena toda la información relacionada con la misma. El servidor procesa la petición y devuelve la respuesta al navegador que la presenta al usuario. Por tanto, el sistema se distribuye en tres componentes: el navegador, que presenta la interfaz al usuario; la aplicación, que se encarga de realizar las operaciones necesarias según las acciones llevadas a cabo por éste y la base de datos, donde la información relacionada con la aplicación se hace persistente. Esta distribución se conoce como el modelo o arquitectura de tres capas.

En la mayoría de los casos, el navegador suele ser un mero presentador de información (modelo de cliente delgado), y no lleva a cabo ningún procesamiento relacionado con la lógica de negocio. No obstante, con la utilización de applets, código javascript y DHTML la mayoría de los sistemas se sitúan en un punto intermedio entre un modelo de cliente delgado y un modelo de cliente grueso (donde el cliente realiza el procesamiento de la información y el servidor sólo es responsable de la administración de datos). No obstante el procesamiento realizado en el cliente suele estar relacionado con aspectos de la interfaz (como ocultar o mostrar secciones de la página en función de determinados eventos) y nunca con la lógica de negocio.

En todos los sistemas de este tipo y ortogonalmente a cada una de las capas de despliegue comentadas, podemos dividir la aplicación en tres áreas o niveles:

- **Nivel de presentación:** es el encargado de generar la interfaz de usuario en función de las acciones llevadas a cabo por el mismo.

- **Nivel de negocio:** contiene toda la lógica que modela los procesos de negocio y es donde se realiza todo el procesamiento necesario para atender a las peticiones del usuario.
- **Nivel de administración de datos:** encargado de hacer persistente toda la información, suministra y almacena información para el nivel de negocio.

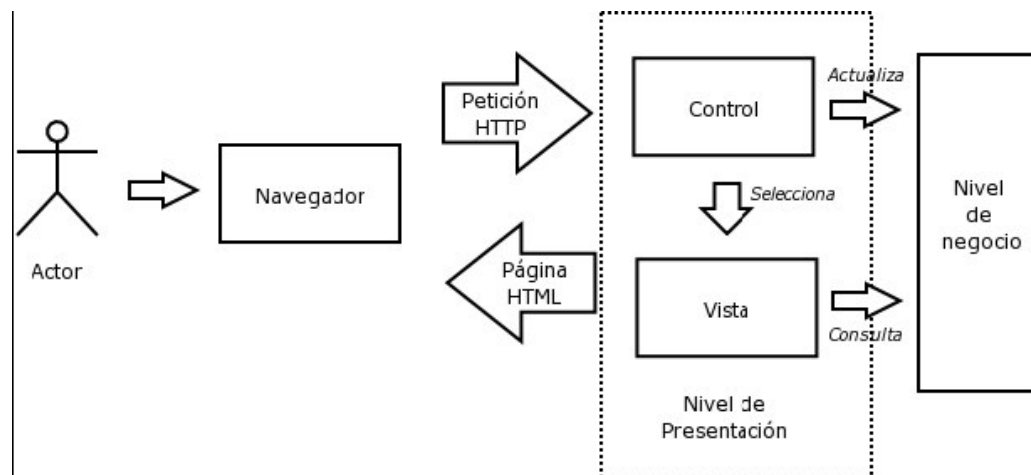
Los dos primeros y una parte del tercero (el código encargado de las actualizaciones y consultas), suelen estar en el servidor mientras que la parte restante del tercer nivel se sitúa en la base de datos (notar que, debido al uso de procedimientos almacenados en la base de datos, una parte del segundo nivel también puede encontrarse en la misma). Teniendo en cuenta estas características en la arquitectura de los sistemas Web, a continuación veremos algunos patrones de diseño de aplicación básica que pueden facilitar un diseño apropiado para este tipo de sistemas.

Uno de los patrones que ha demostrado ser fundamental a la hora de diseñar aplicaciones Web es el **Modelo-Vista-Control (MVC)**. Este patrón propone la separación en distintos componentes de la interfaz de usuario (vistas), el modelo de negocio y la lógica de control. Una vista es una “fotografía” del modelo (o una parte del mismo) en un determinado momento. Un control recibe un evento disparado por el usuario a través de la interfaz, accede al modelo de manera adecuada a la acción realizada, y presenta en una nueva vista el resultado de dicha acción. Por su parte, el modelo consiste en el conjunto de objetos que modelan los procesos de negocio que se realizan a través del sistema.

En una aplicación Web, las vistas serían las páginas HTML que el usuario visualiza en el navegador. A través de estas páginas el usuario interactúa con la aplicación, enviando eventos al servidor a través de peticiones HTTP. En el servidor se encuentra el código de control para estos eventos, que en función del

evento concreto actúa sobre el modelo convenientemente. Los resultados de la acción se devuelven al usuario en forma de página HTML mediante la respuesta HTTP.

Figura 9. Representación del modelo tres capas



La clave está en la separación entre vista y modelo. El modelo suele ser más estable a lo largo del tiempo y menos sujeto a variaciones mientras que las vistas pueden cambiar con frecuencia, ya sea por cambio del medio de presentación (por ejemplo HTML a WAP o a PDF) o por necesidades de usabilidad de la interfaz o simple renovación de la estética de la aplicación. Con esta clara separación las vistas pueden cambiar sin afectar al modelo y viceversa. Los controladores son los encargados de hacer de puente entre ambos, determinando el flujo de salida de la aplicación (qué se ve en cada momento).

2.4 TECNOLOGIA .NET

2.4.1 ¿Que es .NET?. Con el cambio de ideología que actualmente se esta presentando en Microsoft, la cual esta dejando a un lado el desarrollo de aplicaciones propias de código cerrado, para generar herramientas ceñidas a los estándares que le permitirán competir con calidad de servicios y no con la desgastante invención continua de productos, ha llevado a la aparición de nuevas tecnologías tales como Windows XP, Office XP, .NET Framework, la nueva gama de servidores, y el VS.NET que precisamente buscan ofrecer servicios informáticos de alta calidad a través de redes TCP/IP y Web.

Es por ello que Microsoft prácticamente ha rediseñado toda su infraestructura tecnológica para ceñirla a los estándares internacionales. Visual Studio .NET es parte del rompecabezas de esta tecnología en lo que respecta al desarrollo de aplicaciones y servicios que puedan aprovechar las bondades de Internet y redes Web para poder funcionar de mejor forma.

2.4.2 Orígenes e Ideología. La globalización mundial axial como los grandes adelantos tecnológicos que cada vez se van desarrollando en forma acelerada, indican que actualmente no solo se deben desarrollar herramientas informáticas muy sofisticadas, sino que además deben ser muy robustas y compatibles entre si y con otras tecnologías diferentes, es decir que ya no se puede estar aislado del mundo y tratar de monopolizarlo, existen muchas y muy diversas tecnologías, entonces el objetivo consiste en masificar las propias aplicaciones y permitir que estas se puedan comunicar con cualquier otro tipo de tecnología diferente, basándose en los estándares de comunicación existente.

Por lo anterior, Microsoft ha entrado en una era de cambio total de su ideología, lo que le ha llevado a generar herramientas y soluciones multifuncionales que pueden funcionar desde cualquier tipo de sistema operativo, así, lo que sólo podía usarse por quienes tuvieran Windows (como los componentes COM, DCOM o

COM+, así como los ActiveX), ahora mediante SOAP, XML, WSDL, UDDI, y otros, podrá ser utilizado por quien se base en estas tecnologías (sin importar su plataforma). Esto permitirá que se ofrezcan, efectivamente, servicios por Internet, que podrían coadyuvar a generar mejores oportunidades de negocio, ya sea al arrendar tales servicios, o al facilitar el desarrollo de sitios Web que mejoren las capacidades funcionales ofrecidas a los usuarios.

De esta forma, un proveedor de servicios (una empresa de cualquier tamaño) puede valerse de la tecnología .NET para agilizar sus procesos productivos relacionados con Internet, redes Web o aplicaciones de escritorio. Si es usuario, podrá encontrarse con una mayor gama de servicios y más ricos en facultades conforme estas tecnologías tomen mayor fuerza en Internet. Su competitividad como prestador de Servicios Web XML deberá basarse en su calidad y, por ende, confiabilidad. Ciertamente habrá más de un servicio que proporcionará una funcionalidad determinada, pero en su riqueza, calidad y confiabilidad estará basado su éxito.

La meta actual de Microsoft es desarrollar, entregar y masificar herramientas de tecnología informática de alta calidad, con alto grado de conectividad y escalabilidad, que además, como siempre ha sido la política de Microsoft, sean sencillas de manejar y permitan realizar al usuario gran cantidad de tareas sin requerir grandes conocimientos en ciencias informáticas, sin generar esfuerzos y economizando recursos.

Para los desarrolladores, Visual Studio .Net, ofrece una herramienta de gran calidad, con muchas posibilidades de trabajo y brindando facilidades tales como: el manejo de datos a través de controles fácilmente configurables, la creación de clases y objetos utilizando ayudantes o wizards, conectando aplicaciones de manera rápida con solo hacer un clic y sin escribir una sola línea de código; lo que le permite al desarrollador dedicar su tiempo a las etapas importantes de su

aplicación, tales como el análisis y el diseño, para lo cual también VS .Net ofrece ayuda con diferentes herramientas para realizar los diferentes diagramas de diseño.

Además con la aparición del Framework .Net, permite una alta compatibilidad de este sistema pues se pueden diseñar aplicaciones sin importar que tipo de lenguaje de programación se este utilizando, lo cual facilita la programación pues pueden participar profesionales expertos en distintos lenguajes para desarrollar una misma aplicación. El objetivo es cumplir con la máxima de la programación orientada a objetos, que es la reutilización, para aprovechar el uso de código ya hecho y evitar rehacer el trabajo.

Con esto en mente, Microsoft generó un marco de trabajo que fuera aprovechado por cualquier lenguaje de programación que se ajustara a sus estándares. Ese marco de trabajo es el .NET Framework, que es el meollo de la tecnología .NET. En pocas palabras, el .NET Framework es un cúmulo de clases expuestas para que, quien quiera, haga uso de su funcionalidad. A su vez, este cúmulo de clases conforma un estándar abierto que puede integrarse a cualquier plataforma o lenguaje.

Esta apertura permitió el diseño de un entorno de desarrollo tan amplio como lo es el Visual Studio .NET, que no sólo incluye los lenguajes de C# .NET, Visual Basic .NET, Visual C++ .NET, Visual J# .NET, sino que hay más de 20 lenguajes de otros fabricantes que pueden funcionar en él, como Pascal .NET, Cobol .NET, y muchos otros.

Así pues, todo lleva a concluir que Internet y las redes Web ahora sí serán utilizados como un medio para apalancar a las empresas que ya estén establecidas físicamente en algún lugar. Se facilitará el intercambio de información y la agilidad para realizar una aplicación. Se reducirán los tiempos de desarrollo, y

se obtendrían mejores resultados cuando se realice alguna tarea en este entorno. Así pues, Windows XP, Visual Studio .NET, Office XP y toda la gama de tecnologías que Microsoft recién ha presentado y que próximamente presentará se ciñe al nuevo esquema de la tecnología .NET.

2.4.3 Aplicaciones Web con ASP .NET de Visual Studio. Visual Studio .NET permite crear aplicaciones que aprovechan la potencia de World Wide Web. En ello se incluye todo, desde un sitio Web tradicional que sirva páginas HTML hasta completas aplicaciones de negocios que se ejecuten en una intranet o en Internet, sofisticadas aplicaciones que proporcionen componentes basados en Web para el intercambio de datos mediante XML.

- **Dónde es adecuado Visual Studio.** Como con cualquier aplicación .NET, si dispone de .NET Framework, puede crear aplicaciones ASP.NET mediante editores de texto, un compilador de línea de comandos y otras herramientas sencillas. Puede copiar archivos manualmente en IIS para implementar la aplicación.

También se puede utilizar Visual Studio. Cuando utilice Visual Studio para crear aplicaciones Web, estará creando esencialmente la misma aplicación que podría crear a mano. Es decir, Visual Studio no crea un tipo diferente de aplicación Web; el resultado final continúa siendo una aplicación Web ASP.NET.

La ventaja de utilizar Visual Studio reside en que proporciona herramientas que facilitan el desarrollo de aplicaciones y lo hacen más rápido y confiable. Entre estas herramientas se incluyen:

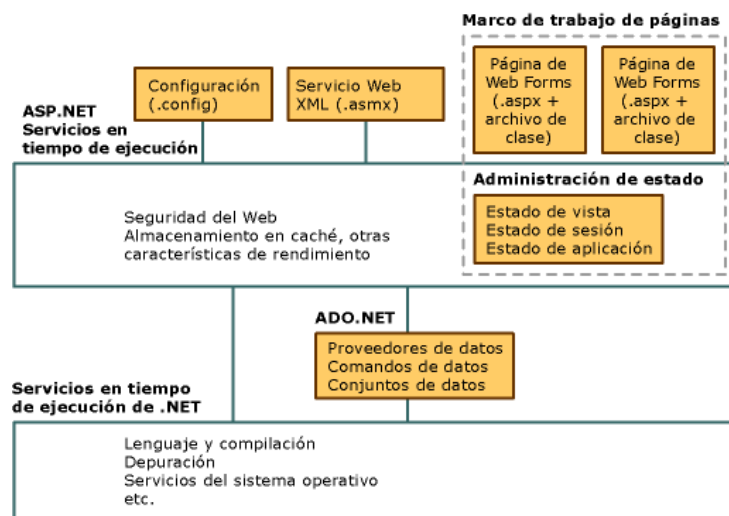
- Diseñadores visuales para páginas Web con controles para arrastrar y colocar, y vistas de código (HTML) con comprobación de sintaxis.

- Editores de código inteligentes que incluyen finalización de instrucciones, comprobación de sintaxis y otras características de IntelliSense.
- Compilación y depuración integradas.
- Utilidades de administración de proyectos para la creación y administración de archivos de aplicación, incluida la implementación en servidores locales o remotos. Si ha utilizado antes Visual Studio, este tipo de características le resultarán familiares, ya que son similares a las que se utilizaban para crear aplicaciones en versiones anteriores de Visual Basic y Visual C++. Con Visual Studio .NET puede utilizar estas funciones para crear aplicaciones Web ASP.NET.
- **Elementos de las aplicaciones Web ASP.NET.** La creación de aplicaciones Web ASP.NET implica trabajar con muchos de los elementos que se utilizan en cualquier aplicación de escritorio o cliente-servidor. Éstos incluyen:
 - **Funciones de administración de proyectos** Al crear una aplicación Web ASP.NET, necesitará hacer un seguimiento de los archivos necesarios, cuáles se debe compilar y cuáles necesitan implementarse.
 - **Interfaz de usuario** Una aplicación suele presentar información a los usuarios; en una aplicación Web ASP.NET, la interfaz de usuario se presenta en páginas de formularios Windows Forms, que envían los resultados a un explorador. Opcionalmente, puede crear resultado adaptado para dispositivos móviles u otros aparatos Web.

- **Componentes** Muchas aplicaciones incluyen elementos reutilizables que contienen código para ejecutar tareas específicas. En las aplicaciones Web, puede crear estos componentes como servicios Web XML, lo que les permite aceptar llamadas de todo el Web desde una aplicación Web, otro servicio Web XML o un formulario Windows Forms, por ejemplo.
- **Datos** La mayoría de las aplicaciones necesitan algún mecanismo de acceso a datos. En las aplicaciones Web ASP.NET, puede utilizar ADO.NET, los servicios de datos que forman parte de .NET Framework.
- **Seguridad, rendimiento y otras características de infraestructura** Como en cualquier aplicación, deberá implementar seguridad para evitar el uso no autorizado, probar y depurar la aplicación, ajustar su rendimiento y ejecutar otras tareas no relacionadas con la función principal de la aplicación.

El diagrama siguiente proporciona información general sobre el modo en que encajan entre sí las diferentes piezas de las aplicaciones Web ASP.NET y sobre cómo encajan en el contexto más amplio de .NET Framework.

Figura 10. Contexto del .NET Framework



- **Administración de proyectos: proyectos de aplicación Web ASP.NET e implementación.** Para trabajar con una aplicación Web ASP.NET utilizando Visual Basic o Visual C#, usará la plantilla de proyecto Aplicación Web ASP.NET. Como sucede con otros proyectos de Visual Studio, un proyecto de aplicación Web es un repositorio central para toda la información necesaria para diseñar, ejecutar y administrar la aplicación.

Cuando se crea un proyecto de aplicación Web, Visual Studio crea los archivos y carpetas necesarios en el servidor, establece sobre ellos las configuraciones de seguridad adecuadas y crea la aplicación IIS.

Cada aplicación Web puede mantener un archivo Web.config que sigue el formato y las convenciones de los archivos de configuración .NET. En ASP.NET, el archivo de configuración permite establecer configuraciones específicas del proyecto para la seguridad, las opciones de compilación, el seguimiento, el control de errores, etc.

- **Interfaz de usuario: los formularios Web Forms y el marco de trabajo de páginas ASP.NET.** Puede crear páginas HTML y páginas de formularios Web Forms como interfaz de usuario para la aplicación. Las páginas HTML se utilizan generalmente para contenido estático. Por el contrario, las páginas de formularios Web Forms proporcionan una interfaz programable que funciona de forma muy parecida a un formulario de Visual Basic, excepto en que la interfaz de usuario se representa en un explorador Web u otro dispositivo Web.

Las páginas de formularios Web Forms se generan en el marco de trabajo de páginas ASP.NET. Cada página de formularios Web Forms es un objeto que se deriva de la clase Page de ASP.NET, que actúa como contenedor para los

controles. La página en realidad consta de dos archivos: un archivo .aspx que contiene los elementos de la interfaz de usuario (texto HTML estático y elementos de control) y un archivo de clase que contiene el código que devuelve la página.

Cuando los usuarios piden una página de formularios Web Forms, el marco de trabajo de páginas ejecuta el objeto página de formularios Web Forms y todos los controles individuales que contiene. El resultado de la clase **Page** y de los controles es HTML.

Dado que el Web es inherentemente dinámico (de forma predeterminada, los componentes de una aplicación Web sólo existen durante el tiempo necesario para procesar una única petición), las aplicaciones Web afrontan un desafío al conservar los valores mientras el usuario trabaja con la aplicación. Como ayuda, el marco de trabajo de páginas incluye utilidades para la administración del estado. Entre ellas se incluye una utilidad "estado de vista" basada en páginas (un método para conservar los valores de los controles) y el acceso a utilidades de estado no basadas en páginas tales como un estado de sesión (específico del usuario) y un estado de la aplicación (global para la aplicación).

El marco de trabajo de páginas admite un conjunto de controles que pueden utilizarse para programar la interacción del usuario con las páginas de formularios Web Forms. Las acciones que el usuario realiza sobre el formulario son capturadas y procesadas por el marco de trabajo de páginas de forma que puedan tratarse como eventos estándar. Puede elegir entre una amplia selección de controles disponibles en Visual Studio. Además, puede crear controles personalizados propios. Nota También puede crear resultados para dispositivos móviles. Para hacerlo así, se utiliza el mismo marco de trabajo de páginas ASP.NET, pero se crean formularios Mobile Web Forms en lugar de páginas de formularios Web Forms y se utilizan controles diseñados específicamente para dispositivos portátiles.

- **Componentes basados en Web: servicios Web XML.** Un servicio Web XML es un componente al que otras aplicaciones pueden llamar a través de una red TCP/IP. Ejecuta una función específica (cualquier cosa desde cálculos y validación de tarjetas de crédito hasta procesamiento de pedidos complejos) y devuelve valores a la aplicación que hace la llamada.

Lo que hace únicos a los servicios Web XML es que se les puede llamar a través del Web. Los servicios Web XML se invocan mediante peticiones HTTP o SOAP e intercambian datos con otros componentes mediante XML. Como tales, pueden convertirse en parte integral de aplicaciones Web ASP.NET y proporcionar servicios no sólo a las aplicaciones, sino también a cualquier aplicación que disponga de acceso Web, por lo que resultan ideales para las transacciones de negocio a negocio.

- **Acceso a datos en aplicaciones Web.** La mayoría de las aplicaciones Web ASP.NET implican al menos un cierto nivel de acceso a datos. ASP.NET no incluye directamente posibilidades de acceso a datos. En su lugar, las aplicaciones Web utilizan servicios de datos ADO.NET.

ADO.NET proporciona un marco de trabajo completo para obtener acceso a datos de distintos orígenes y administrarlos, incluidas las bases de datos y los archivos XML o las secuencias. ADO.NET incluye proveedores: clases que permiten conectar con orígenes de datos, ejecutar comandos y leer resultados. De modo opcional, puede conservar datos en un conjunto de datos, que es una caché en memoria desconectada.

El acceso a datos en las aplicaciones Web, ya sea en una página de formularios Web Forms o en un servicio Web XML, supone desafíos especiales:

- **Dinamismo** Los componentes de las aplicaciones Web no suelen conservar un estado, lo que hace que no sea práctico mantener conexiones activas con un origen de datos (u otros recursos).

- **Escalabilidad** Dado que las aplicaciones Web pueden tener cargas de usuario que varían sustancialmente en cortos periodos de tiempo, el acceso a los datos debe diseñarse teniendo en cuenta la escalabilidad.

- Visual Studio proporciona diversas herramientas para trabajar con datos, incluida la compatibilidad del Cuadro de herramientas con los elementos de datos, diferentes asistentes para configuración, etc.

- **Infraestructura de las aplicaciones Web: seguridad, rendimiento y otros aspectos.** Además de facilitar el medio para crear elementos de interfaz de usuario y componentes a los que se puede llamar, ASP.NET proporciona un contexto para ejecutar esos elementos. Por ejemplo, ASP.NET se comunica con IIS para controlar solicitudes de páginas de formularios Web Forms y servicios Web XML, analizar los archivos, llamar a componentes relacionados, etc.

Gran parte de este trabajo tiene lugar bajo la superficie, en un nivel en el que habitualmente no es necesario programar durante la creación de aplicaciones de escritorio.

Visual Studio proporciona una compatibilidad limitada para trabajar con las características de infraestructura de ASP.NET. Por ejemplo, puede editar el archivo de configuración de la aplicación Web (Web.config) mediante el Editor de código de Visual Studio. No obstante, .NET Framework es tanto conectable como extensible y ofrece acceso de bajo nivel, si es necesario.

Hay otros aspectos de las aplicaciones Web con los que deberá enfrentarse que forman parte de la infraestructura de la aplicación. Éstos incluyen:

Seguridad Con frecuencia, deberá autenticar y autorizar a los usuarios de la aplicación Web. Hay problemas especiales asociados con la seguridad en una aplicación Web, porque los usuarios tienen acceso a recursos basados en servidor y porque se tiene muy poco control sobre el lado cliente de la aplicación (el explorador o el dispositivo portátil). ASP.NET incluye funciones de seguridad que pueden configurarse y programarse en la aplicación Web.

Rendimiento y optimización Para ajustar el rendimiento de la aplicación, puede utilizar memoria caché de páginas y datos. ASP.NET mantiene una caché de resultados que almacena páginas pedidas anteriormente; al especificar configuraciones de caché, es posible controlar cuánto tiempo se mantienen las páginas en la memoria caché y bajo qué circunstancias se actualizan.

Seguimiento Dado que las aplicaciones Web se ejecutan en el servidor, a menudo un servidor remoto, no tienen otro resultado que el resultado de la aplicación (una página de formularios Web Forms, por ejemplo). ASP.NET, en consecuencia, ofrece la oportunidad de incluir información de seguimiento directamente en una página de formularios Web Forms.

3. DESARROLLO DE LA PRÁCTICA

En el presente capítulo plantea el contexto y las actividades realizadas durante los seis meses de duración de la práctica empresarial, que permitieron cumplir con los objetivos a desarrollar, y describe el apoyo brindado a las diferentes etapas de los proyectos que en el momento se venían realizando dentro de la empresa; haciendo énfasis en que no se tenía asignado ningún proyecto específico, sino que se realizaban labores que permitieran optimizar los procesos para la construcción de los mismos.

Inicialmente se revisó la documentación existente y la fase de evolución de los proyectos que se estaban realizando en el momento, sobretodo de la aplicación principal que se estaba desarrollando, la cual consiste en el desarrollo de un sistema de información en ambiente Web para el manejo y venta de servicios funerarios para un parque cementerio que permita el manejo de inventario, facturación, cartera y comisiones; que fue el motivo fundamental para trabajar con herramientas Visual Studio .Net 2005.

Cabe resaltar la magnitud e importancia de un proyecto de este tipo ya que se pretende manejar en ambiente Web un sistema de información donde la cantidad de transacciones es bastante grande y se maneja procesos que son de vital importancia para la empresa cliente; lo cual implica un especial cuidado con el manejo y la seguridad de los datos pues la proyección del software es que funcione tanto en la intranet como desde Internet.

Con este proyecto se debe destacar la visión ambiciosa de la empresa cliente y de ISL Ingenieros de Sistemas Ltda. Siendo pioneros en la región, en la realización de un sistema de las magnitudes del que se pretende desarrollar y en

ambiente Web, utilizando ultimas tecnologías tales como SQL Server 2005 y Visual Studio .Net 2005 entre otras y convirtiendo a la empresa cliente en una de las pocas que posee un sistema de información completo para el manejo de sus procesos en Internet.

3.1 DESCRIPCION DE LA APLICACIÓN

La aplicación consta de cuatro módulos que constituyen el manejo de los procesos fundamentales para el desarrollo de las actividades de esta empresa, ellos son:

3.1.1 Modulo de inventario. Debido a las características especiales de los productos ofrecidos por el parque cementerio y los servicios asociados a los mismos, este modulo permite manejar el histórico de ventas y ocupaciones, además de integrar el inventario a los demás procesos de la empresa debido a que actualmente esta información esta consignada en libros y hace difícil el manejo de la situación actual del parque.

3.1.2 Modulo de Cartera. En este modulo permite manejar las cuentas y facturas por cobrar así como los intereses aplicados y las formas de financiación, debido a lo especial de los productos ofrecidos por la empresa, se realizan normalmente ventas a crédito.

3.1.3 Modulo de facturación. Este modulo se encarga de generar las distintas facturas de venta de los diferentes productos y servicios ofrecidos por la empresa, así mismo es la encargada de alimentar el sistema de comisiones por venta de los diferentes asesores

3.1.4 Modulo de Comisiones Nomina. Este modulo se encarga de realizar la liquidación mensual de las comisiones por venta para los asesores de acuerdo los

tipos de productos vendidos, la modalidad de pago y una tabla de porcentajes establecida por las directivas de la empresa.

Estos cuatro módulos deben complementarse entre sí, para permitir una correcta interacción entre los procesos de la empresa; por lo tanto, con la importancia de estos procesos para la organización, el cliente decidió utilizar herramientas de última tecnología que combinadas con un excelente desarrollo brindado por ISL, le permitan obtener un software exclusivo de alta calidad; por tal motivo y por efectos de las cláusulas de confidencialidad no fue autorizada la reproducción parcial de imágenes, código fuente o diagramas de diseño, ya que todo esto es propiedad de la empresa cliente.

3.2 ANALISIS Y DISEÑO DE LA APLICACIÓN

El principal problema que detectado fue adaptar las nuevas metodologías de desarrollo software a la que actualmente ha venido manejando la empresa, y que ha utilizado para la realización de proyectos anteriores; tratando de minimizar el impacto y que no implique desechar el trabajo realizado hasta el momento dentro de la aplicación.

En esta etapa del proyecto fueron repartidos algunos roles dentro del proyecto, adicionales a las labores de desarrollo y construcción de la aplicación, tales como el manejador de la base de datos (DBA), el diseñador de reportes e informes, consultor de nuevas tecnologías.

Para el software diseñador de reportes se tenían dos opciones para elegir: la primera el SQL Reporting Services de SQL Server 2005, y el diseñador Crystal Reports 10 incluido dentro de la herramienta Visual Studio .NET 2005; se

recomendó Crystal Reports porque brinda mayor facilidad para diseño de reportes y permite una mejor estructuración; además que se tiene mayor experiencia en esta herramienta pues en los desarrollos anteriores de la empresa se han utilizado versiones de este software.

El principal problema que se presenta con Crystal Reports fue de conexión con la base de datos, pues con un sistema distribuido Web cliente servidor se debe manejar mucha seguridad en la base de datos, y el problema se presenta cuando al cargar el reporte desde un cliente, la base solicitaba contraseña de entrada y no permite cargar el reporte. Este problema se soluciona utilizando una conexión SQL Native Client que permite acceder al reporte desde cualquier computador sin requerir la contraseña de la base de datos y controlando al mismo tiempo la seguridad ya que solo permite acceder a los usuarios del sistema.

3.2.1 Estudio y refinación del los requerimientos del proyecto. En primer lugar se revisó toda la documentación existente en cuanto al análisis y especificación de requerimientos, para apropiarnos de ellos y empezar el trabajo de diseño de la aplicación el objetivo era el de clarificar las dudas con respecto a todos los requisitos del sistema y la definición del orden de construcción de los diferentes módulos así como darnos una idea de la definición de las reglas del negocio.

Se revisaron los mapas de procesos actuales de la empresa cliente para definir cuales de ellos se consideraban adecuados y eficientes, para con ellos definir las reglas del negocio. Así como también se revisaron la adopción de las tecnologías .Net haciendo que ISL reconfigurara sus estándares y procedimientos de producción de software para que estos se adaptaran a las ventajas que proponen las nuevas tecnologías.

Uno de los principales problemas que se presenta con un sistema de información orientado a Web es la seguridad de la información interna de la empresa; mas aun si el objetivo a mediano plazo no es solo trabajar en la intranet sino también tener la facilidad de trabajar el sistema desde Internet en cualquier parte del mundo, con lo cual se debe plantear especial atención y cuidado en el manejo de datos entre paginas de la aplicación; entre el servidor y el cliente, etc. para evitar que se filtre información fuera de la organización con las consecuencias que esto conlleva o que permita a personas ajenas acceder a información privada de la empresa.

3.2.2 Apoyo a la definición y documentación del diseño de la aplicación. En esta etapa se revisaron y desarrollaron todos los procedimientos especificados para el cumplimiento de la etapa de diseño, se realizaron algunas adaptaciones del modelo actual que permitieran un uso optimo de las herramientas tecnológicas seleccionadas tales como el uso de diagramas de clase lo que permite una visión mas global del sistema, tratando de entregar al final una documentación que permita la construcción de la aplicación sin que se presenten mayores contratiempos.

- **Esquematzación y descripción de las reglas del negocio (segunda capa).** Con la apropiación de los requerimientos y la revisión de los procedimientos se continuó con la esquematización de los principales procesos de los que se va a encargar el sistema, sus relaciones entre sí y la interacción de cada uno de ellos con los usuarios del sistema; se inicia la documentación de la etapa de diseño y la preparación del informe que se le debe entregar al cliente para la respectiva aprobación de la fase, dicho documento contiene entre otros los siguientes:

Modelo actualizado del negocio.

Árbol de menús.

Diseño de reportes y pantallas.

Diseño detallado de almacenamiento de datos.
Especificaciones de comunicaciones.
Especificaciones de respaldo y recuperación.
Requerimientos seleccionados y diseño básico del primer prototipo.
Especificaciones de programación.
Plan de contingencia.
Plan de pruebas del software.⁶

En esta fase se definieron diferentes puntos sobre la presentación del sistema en cuanto a lo que tiene que ver con la combinación de colores a utilizar, diseño de pantallas manejo de logotipos; se propuso el nombre del software (STIGIA – Sistema de Tradicionales) y se realiza una presentación por parte del director técnico del proyecto ante las directivas del cliente para la aprobación de estos ítems.

- **Definición de políticas de seguridad, roles y niveles usuarios.** Se revisaron las diferentes estrategias para la definición de esquemas de seguridad, así como el manejo de roles y usuarios dentro de la aplicación, y debido a la importancia de la información manejada dentro del sistema y a la escalabilidad del mismo, donde se pretende que se pueda trabajar desde cualquier lugar utilizando Internet.

Debido a esto y por solicitud del cliente no solo se utilizaron los procesos de autenticación de formularios y permisos de usuario que automáticamente maneja ASP .NET, sino que además se requiere que el sistema autentique contra el Directorio Activo del Servidor, lo cual implica el manejo de distintas variables de entorno y la incorporación de métodos y procedimientos mas confiables de control y auditoria de procesos. De acuerdo con estas premisas se definieron 5 niveles o roles de usuario así:

⁶ Tomado de *PTPS01 PROCESO PRODUCCIÓN DE SOFTWARE*. Documento Interno. ISL.

- *Administrador General:* este rol tiene la capacidad de manejar todos los procesos del sistema, así como la creación de usuarios y la definición de permisos para los mismos, esta encargado de la alimentación de las tablas básicas, se encarga de la verificación y auditoria de los procesos y verificación del correcto funcionamiento del mismo
- *Administrador de Modulo:* tiene a cargo todos los procesos de su área y posee todos los permisos para la realización de operaciones que tengan que ver con el modulo asignado a la dependencia; los administradores de modulo serán por lo general jefes del departamento, pues tienen autorización para tomar decisiones sobre procesos y procedimientos a su cargo
- *Usuario del Modulo:* permite la consulta, generación, modificación y ejecución de procesos asignados a su área
- *Usuario Consulta del Modulo:* este rol permite al usuario cuyo rol pertenezca a un modulo diferente consultar reportes y resultados de procesos generados por otros módulos.
- *Usuario Visitante:* usuario al que no se le permite realizar procesos de ningún modulo pero que tiene acceso a algunas utilidades de la aplicación y consultar información de la misma

El sistema de seguridad ofrece la ventaja adicional de que si hay algún tipo de proceso especial este puede asignarse a un usuario específico, además un usuario puede llegar de acuerdo a los alcances de su cargo a tener varios roles dentro de los diferentes módulos del sistema.

También se tiene que tener en cuenta que si un empleado es retirado de su cargo y se retira su usuario del Active Directory, este ya no podrá volver a acceder al

sistema, pero la información existente asociada a este usuario permanecerá en el sistema sin generar ningún tipo de conflicto.

- **Definición de la estructura de datos.** Esta etapa define sobre todo la estructura de los datos que se van a manejar dentro del sistema, su control y la seguridad de los mismos dentro de la aplicación, se tenía especial cuidado en el rendimiento, la optimización y escalabilidad de la aplicación, se conoce de antemano la gran cantidad de información a manejar y el constante flujo de datos dentro del sistema.

El principal objetivo entonces era evitar que por la gran cantidad de datos a manejar se generen demoras en la carga de la información para el usuario, retrasos en la ejecución de procesos o pérdida de información relevante del sistema lo que perjudicaría la confiabilidad del software dentro de la organización

- **Diseño, documentación y construcción de la base de datos del sistema.** Teniendo en cuenta la definición de la estructura de datos especificada anteriormente se inicia el diseño de la base de datos utilizando el motor de base de datos SQL Server 2005 previamente aprobada por el cliente, además se hace entrega de la administración de la base al DBA (Administrador de Base de Datos) asignado por la dirección técnica del proyecto, quien se encarga de crear los niveles de usuario de la base, definir la viabilidad de cambios solicitados, así como de mantener informado al equipo de desarrollo de todos los cambios que se realicen a la base; además de mantener actualizada la información concerniente con la documentación de la base de datos entre los que se encuentran los diagramas de entidad – relación y estructuras de datos entre otros.

3.3 CONSTRUCCION DE LA APLICACIÓN

Con base en el diseño realizado en la fase anterior y utilizando la metodología de prototipado que se seleccionó con anterioridad, se inicia la fase de construcción de la aplicación.

Se pretende construir un par de prototipos por cada modulo iniciando por el modulo de inventario, se realizaran los ajustes necesarios y luego se inicia la construcción del prototipo del siguiente modulo, que se ira acoplando al anterior y se realizaran los ajustes necesarios, el objetivo es generar el acoplamiento en la medida que se va desarrollando, hasta lograr un ensamble total de todo el sistema con sus diversos módulos totalmente terminados.

Se podrán realizar ajustes de los módulos anteriores en cualquier fase del proceso de acoplamiento lo que nos permitirá refinar con cada paso todos los requerimientos y corregir fallas que se vayan presentando en los módulos anteriores; hay que mencionar que debido a la necesidad y premura para las entregas se definió por parte de las directivas unificar esfuerzos y dedicar toda la fuerza del grupo de desarrollo a un solo modulo a la vez, por lo tanto no se continua con comisiones sino que se inicia el apoyo al modulo de facturación, que no se tenia contemplado.

Se configuró el entorno de desarrollo y la división de funcionalidades, se seleccionaron los equipos que de acuerdo con su capacidad operativa y su carga de trabajo pudieran desarrollar funciones predeterminadas y adicionales, así se configuraron el servidor de bases de datos, el servidor de aplicaciones y el repositorio de datos

El servidor de aplicaciones se encarga de brindar los servicios de acceso a la aplicación desde cualquier punto de la red local mediante el protocolo TCP/IP.

Para esto se configuró dentro de los servicios de IIS (Internet Information Server) una carpeta virtual y los parámetros de funcionamiento de ASP .NET para lo cual tuvo que ser instalado el Marco de trabajo de Microsoft: NET Framework en su versión 2.0.

El servidor de base de datos contiene la base de datos y gestiona el acceso de la aplicación a los mismos. En este equipo debe instalarse SQL Server 2005 y, al igual que en el servidor de aplicaciones, el Framework 2.0 este equipo cuenta además con la herramienta: SQL Server Management Studio, la cual permite administrar y configurar todos los servicios de SQL Server además de tareas de creación, respaldo y recuperación de datos.

El servidor de la empresa desarrolladora sirvió para mantener centralizados los archivos del proyecto facilitando el trabajo en equipo y su conservación; ya que el servidor tiene un estricto plan de mantenimiento entre los cuales se cuenta con un Backup semanal que garantiza que ante cualquier eventualidad se puede tener una última versión del proyecto.

3.3.1 Diseño y construcción del primer prototipo. Basados en los documentos realizados en la fase de diseño, diagramas de clases y la especificación de requisitos; se toman los concernientes con el modulo de inventario y se inicia la construcción del primer prototipo, se revisan los procedimientos definidos por el usuario, las relaciones entre sí, se analizan las opciones de desarrollo de estos y la importancia dentro del prototipo, se dividieron las tablas asociadas al inventario entre básicas y maestras, generando las respectivas clases

- **Construcción del módulo 1.** Se definieron las clases que componen este modulo, dando inicio a su construcción utilizando el diseñador de clases de Visual Studio .Net 2005, se inicio la programación de los procesos de inventario

utilizando Visual Basic .Net, se crearon las diferentes pantallas, basándonos en el diseño de la interfaz grafica definida anteriormente y aprobada por el usuario y se inicio la generación de reportes.

Se desarrollaron los métodos y propiedades de cada una de las clases que componen el inventario, las funciones que debe realizar el modulo, así como se definieron la cantidad de reportes que el sistema debe generar y como debe estar organizada la información dentro de los mismos

- **Implementación del modulo en ambiente de prueba.** Se realizaron entregas parciales a la empresa cliente, se instalaron los componentes necesarios para el buen funcionamiento de la aplicación dentro de la organización, además de revisar que todos los equipos a utilizar la herramienta, cumplieran con los requisitos mínimos de funcionamiento y permitieran el buen funcionamiento de la misma

En colaboración con el departamento de sistemas se inicio la población con datos reales para comprobar el buen funcionamiento y la escalabilidad del sistema, y realizando una revisión completa de todas las funciones para verificar que si están cumpliendo con los requisitos expresados por el cliente y revisando que el rendimiento de la aplicación sea optimo con el manejo de un gran volumen de datos.

Se presentaron durante la implantación, algunas fallas que se pudieron rectificar y corregir antes de iniciar el siguiente modulo; el cliente expreso algunas dudas y mostró algunos puntos en los cuales el sistema debería tratar la información de manera diferente, se acataron las solicitudes del cliente debido a que no se salen del análisis de requerimientos y además no comprometían el cronograma de trabajo pues no generaban mayores retrasos.

- **Ajuste de requerimientos y refinación del prototipo.** Debido a que la empresa cliente sufrió algunos cambios en su estructura organizacional, surgieron algunas modificaciones que propiciaron el cambio de algunos requerimientos existentes o el surgimiento de algunos nuevos, los cuales serán tenidos en cuenta durante la siguiente etapa.

Estos cambios dentro de la empresa cliente generaron un replanteamiento en el cronograma de trabajo y en la asignación de tareas dentro del equipo de desarrollo, en primer lugar se invirtió el orden de construcción de los módulos de cartera y facturación, haciendo que todo el equipo se dedicara por completo al desarrollo del módulo de facturación, además paralelamente se realizaban los cambios solicitados al módulo de inventario, adaptando la base de datos a los nuevos requerimientos, y evitando un mayor impacto dentro de la estructura de datos.

3.3.2 Diseño y construcción del Segundo prototipo. Debido a las políticas de la nueva administración, fue solicitada la inclusión dentro del sistema de un nuevo módulo que permitiera apoyar los procesos de gestión comercial de la organización, esto implicó un retraso considerable dentro del cronograma de trabajo, debido a que se deben realizar nuevamente fases del proyecto que ya habían sido concluidas lo que generó una desconcentración por parte del equipo de desarrollo que debió asumir la carga de los nuevos procesos.

- **Construcción del módulo 2.** Ya con los retrasos previamente mencionados se inicio el desarrollo del módulo de facturación en el cual se estuvo a cargo de la clase Negocio con algunos procesos auxiliares, además del diseño, diagramación, realización y configuración de los reportes tanto de inventario como de facturación.

El equipo de desarrollo se vio debilitado debido al retiro de uno de sus miembros, esto genero un retraso adicional pues se sobrecargo de trabajo a los

desarrolladores restantes y también se invirtió una gran cantidad de tiempo en capacitación y acoplamiento del nuevo miembro al grupo de trabajo, se generaron nuevas prorrogas al cronograma de trabajo original

- **Implementación del modulo en ambiente de prueba.** Se documentaron los procesos realizados hasta este momento y se colaboró con la puesta en marcha de este modulo en ambiente de pruebas dentro del la organización.

Se inicio la construcción de la documentación técnica de los procesos desarrollados hasta el momento basados en los estándares específicos que tiene ISL para el desarrollo de este tipo de manuales, se supone que estos están sujetos a cambios en la medida que se realicen los módulos restantes y serán terminados por las personas que reemplacen al estudiante en practica.

3.3.3 Diseño y Construcción del Tercer y Cuarto Prototipo. Ya en este punto se acerca el final de la fase de ejecución del cronograma establecido dentro de la práctica empresarial, se inicia el proceso de documentación y revisión de procedimientos para realizar la entrega del cargo a las nuevas personas encargadas de continuar con el proyecto, tratando de capacitarlas lo mejor posible para que no se generen contratiempos con la ejecución del mismo.

Se debe destacar que aunque la aplicación no fue terminada en su totalidad, se cumplió casi en su totalidad con el cronograma establecido dentro del plan de práctica, así como también se desarrollaron los objetivos de la misma, los cuales pretendían apoyar el desarrollo de los diferentes procesos para la utilización de nuevas tecnologías en los proyectos informáticos de la empresa.

Se debe resaltar que los retrasos generados durante la práctica que impidieron el desarrollo total de la misma, estaban fuera de control por parte del estudiante en

práctica y que se debe tener en cuenta que este tipo de contratiempos son normales en el diario vivir de una empresa de desarrollo de software y que por lo tanto hacen parte de la cotidianidad del ingeniero de sistemas.

- **Construcción del modulo 4 (Comisiones).** Para este modulo se definieron en acuerdo con el cliente los procedimientos básicos para la liquidación mensual de las comisiones de los asesores de ventas, basados en los tipo de productos y en el tipo de pago en el cual se haya concretado el negocio. Así también se identificaron los procesos críticos y los puntos neurálgicos dentro del manejo de las comisiones,

Aunque no se culmino la construcción de este modulo, se realizaron actividades que pretendían apoyar y soportar la construcción de estos módulos, en concordancia con lo estipulado dentro de los objetivos de la práctica, que no llevan a la construcción de ninguna aplicación.

4. DESCRIPCION DE OTRAS ACTIVIDADES Y RESPONSABILIDADES A CARGO

Este capítulo muestra las actividades adicionales que se realizaron durante el periodo de practica, pues para ISL, aunque fuéramos practicantes, teníamos que ocupar un cargo existente dentro de la organización y cumplir con el manual de responsabilidades adjunto al mismo, es decir que a las actividades de apoyo descritas en la practica, se suman actividades tales como soporte a usuario, Help Desk, entre otras, además de cumplir responsabilidades que impliquen el mejoramiento continuo y la calidad, debido a que son una empresa certificada y pretende mantener la certificación.

4.1 ACTIVIDADES DE SOPORTE

Entre las actividades adicionales que se realizaban en la institución durante el periodo de la practica se encuentran actividades de soporte a usuario que implican la atención de clientes de ISL quienes utilizan otros sistemas y servicios que ofrece la empresa y que por algún motivo necesitaban apoyo técnico, bien sea por problemas funcionales, de capacitación, o errores dentro del mismo software.

4.1.1 Apoyo al soporte a usuarios de los sistemas existentes dentro de la organización. Dentro de su paquete de servicios ISL ofrece el servicio de soporte técnico, el cual consiste en realizar el mantenimiento de toda la estructura informática de la empresa cliente, para garantizar su correcto funcionamiento, y

solucionar problemas de manera inmediata, para evitar contratiempos dentro de la organización.

Durante el periodo de práctica fueron asignados dos clientes para prestar este tipo de servicio, y las actividades que se realizaban consistían en:

- **Mantenimiento preventivo de la plataforma informática.** a estas empresas ISL les instalo diversas herramientas software desarrolladas dentro de la empresa, para realizar procesos de la organización cliente, tales como Cartera, facturación, inventario, contabilidad, el objetivo es mantener este software funcionando correctamente, revisando cualquier falla tanto estructural como funcional, además de mantener estos sistemas en un estado optimo de rendimiento, depurando la base de datos y realizando Back Up de la información para evitar algún contratiempo
- *Administración de la red interna de la organización:* se realizan tarea como mantenimiento preventivo de los servidores, revisión periódica de la estructura física de la red, instalación y actualización de antivirus, revisión y mantenimiento periódico de los equipos cliente dentro de la red, esto es de vital importancia debido a que el software instalado funciona por red interna y una falla de la red implicaría que toda la organización quedara detenida.
- *Solución de problemas del cliente:* consiste en solucionar cualquier tipo de problema funcional que se presenten con los software propiedad de ISL además de encontrar alternativas para errores de información cometidos por el usuario, y aunque no son responsabilidad de ISL, se debe ofrecer una alternativa viable para corregir el problema

4.1.2 Instalación y configuración de hardware y software. Para algunos clientes de ISL, especialmente aquellos que tienen contratos de otro tipo con la organización, se les brinda el servicio de instalación de equipos de cómputo, impresoras, servidores etc. La configuración de los mismos para su correcto funcionamiento, además de la instalación del software que el cliente considere necesario para el desarrollo de sus procesos, aclarando por su puesto que todo software y hardware este debidamente licenciado y con autorización para usarse.

4.1.3 Mantenimiento de los diferentes sistemas de la empresa. Cuando se detectan fallas o se presentan nuevos requerimientos en el software desarrollado por la empresa ya sea por parte del cliente, o internamente, se deben realizar todas las correcciones necesarias para solucionar el problema. Además se debe acordar con los clientes usuario de cada sistema un plan para la actualización del software donde se entregan ejecutables con el problema solucionado

Durante la práctica fueron asignados algunos sistemas de propiedad de ISL para realizar su mantenimiento y las reparaciones necesarias a los mismos en caso de aparición de errores. Se tenían encargados los siguientes sistemas:

- **SINCO:** sistema integrado de contabilidad, que consiste en un software desarrollado en Visual Studio 6.0 y que integra o maneja por separado, de acuerdo al gusto del cliente, contabilidad, inventario, facturación y cartera en un solo paquete, aunque también funcionan individualmente.
- **SICME:** sistema integrado para la administración de centro medico y laboratorio clínico, y sus dos versiones modificadas, LAB – MANAGER y OD – MANAGER que apoya todos los procesos asociados al funcionamiento de un laboratorio clínico.

- **REQTEC:** sistema que permite manejar los procesos de asignación de servicios técnicos en una empresa mantenimientos y soporte, tal como una que realice mantenimientos de equipo electrónicos.
- **SABA:** sistema que permite llevar y organizar toda la información concerniente a la gestión humana dentro de una organización, como llevar el seguimiento de la hoja de vida de los empleados.

Al software anteriormente mencionado se les realizaba mantenimiento y seguimiento continuo para evitar la aparición de errores y la pérdida de información importante dentro de las organizaciones de los clientes

4.1.4 Apoyo al desarrollo software de aplicaciones existentes. En algunos casos, los clientes solicitan a ISL el desarrollo de módulos adicionales al software que han adquirido de acuerdo a necesidades particulares de la organización, dependiendo de la complejidad y el impacto de estos módulos dentro del software se analiza la viabilidad de las modificaciones y se asigna los recursos y el equipo de trabajo para la realización de estos anexos, normalmente no se asignan mas de dos personas para trabajar en esto.

Durante este periodo se realizaron algunos módulos de este tipo solicitados por los clientes, por ejemplo se realizo en Visual Studio .Net un modulo que utilizara la base de datos actual del cliente y que permita el control de amonestaciones para sus empleados.

También se realizó en Visual Studio 6.0 un modulo que permitiera mostrar el consolidado de citas de un paciente con la respectiva información de cada cita para revisar el comportamiento del paciente.

Este tipo de desarrollos no implican mucho tiempo y se realizan de manera rápida alternando con las labores fuertes de programación dentro de la empresa.

4.2 ACTIVIDADES DE HELP DESK

El Help Desk consiste en brindar una solución rápida de problemas que se presentan a los clientes y que no implican errores en el software o fallas graves en los equipos, para lo cual se le entregan instrucciones fáciles de entender para el usuario y que permitan solucionar el problema. Si con estas instrucciones el usuario no puede resolver el problema o este es mas grave se realiza la programación de la visita por parte de un miembro del equipo de soporte para que corrija la falla personalmente.

4.2.1 Recepción de solicitudes y problemas reportados por los usuarios.

Consiste en recibir las solicitudes y problemas que reportan los usuarios revisar la gravedad del los mismos y planificar la manera de solucionarlos, en el caso de problemas sencillos de funcionamiento en los cuales el usuario puede fácilmente corregir el problema se brinda apoyo telefónico donde se guía al cliente paso a paso para la solución de el problema, se realiza el registro del mismo en el software destinando para eso dentro de ISL y se documenta la solución

4.2.2 Diagnostico preeliminar de problemas y registro de seguimiento a la

solución de los mismos. En el caso de problemas mas serios se realiza un diagnostico preeliminar, se le informan al cliente la medidas que se van a tomar y el tiempo aproximado que tardara la corrección del problema, se registra la solicitud dentro del software destinado por ISL para eso, se asigna a la persona que se va a encargar de la solución del problema y se realiza seguimiento para verificar que el problema ha sido solucionado satisfactoriamente y que además el

cliente quedo satisfecho con la atención brindada y que de verdad se hayan solucionado los inconvenientes presentados.

4.3 ACTIVIDADES DE CALIDAD

Consisten en todas las actividades tendientes al mejoramiento continuo de la organización y que permiten mantener y mejorar la calidad de los procesos y por ende los servicios prestados por la organización

4.3.1 Apoyo al mejoramiento continuo de la organización desde los procesos a cargo. Busca que a través del cargo desempeñado se detecten y registren en el sistema, posibles fallas o mejoras en todos los procesos con el fin de que se puedan corregir e implementar para evitar fallas que se vean reflejadas en los clientes.

5. CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- El modelo de práctica empresarial es una gran experiencia para el estudiante que adquiere conocimientos en el área laboral, conoce el funcionamiento de una empresa, y las políticas de calidad implementadas para satisfacer a los clientes.
- La utilización de metodologías avanzadas de desarrollo software, así como el uso de herramientas y lenguajes de diseño unificados (UML), facilitan las labores en las diferentes etapas de realización de un proyecto y reduce el tiempo de ejecución, que implica reducción en los costos y cumplimiento de los cronogramas de actividades.
- Encargarse de actividades que implican el contacto con clientes de la empresa, desarrolla en el estudiante, habilidades en relaciones publicas, muy importantes dentro del medio laboral y que es un área que el ingeniero de sistemas mantiene bastante descuidada
- La tecnología .Net es una herramienta formidable para la construcción de aplicaciones software, debido a que facilita las labores de programación, se acopla a las metodologías de desarrollo de aplicaciones actuales, y el soporte que brinda la gran cantidad de documentación existente, facilita la solución de problemas
- SQL Server 2005 es una aplicación excelente para el manejo de datos, pues brinda mucho más que un motor de bases de datos, y con su acople con Visual

Studio .Net 2005, facilitan el manejo de los datos dentro de una aplicación y garantizan su seguridad.

5.2 RECOMENDACIONES

- Tener especial cuidado con la instalación de las herramientas de Visual Studio .Net 2005, así como el acoplamiento con SQL Server 2005, debido a que una correcta configuración, garantiza el aprovechamiento de todas las ventajas que ofrecen
- Modernizar los procesos actuales de producción de software, para que se adapten a las necesidades actuales de los nuevos proyectos, se recomendaría el uso de la metodología RUP para proyectos largos y eXtreme Programming para proyectos cortos.

BIBLIOGRAFIA

BALENA Francesco, Programación Avanzada Con Microsoft Visual Basic.NET. McGraw-Hill Interamericana, 2003.

BUHLER, Erich R. Microsoft Visual Basic .NET: Guía de migración y actualización. McGraw-Hill, 2002.

DAWSON, Christian W. y MARTIN GUETGLÁS, Gregorio. El proyecto fin de carrera en Ingeniería Informática: Una guía para el estudiante. Pearson Ecuación, S.A. Madrid.2002

DOBSON, Rick. Programación de Microsoft SQL Server 2000 con Microsoft Visual Basic.NET. McGraw-Hill, 2002.

DUTHIE, Andrew. Aprenda Ya Microsoft ASP.NET. McGraw-Hill, 2002.

GÓMEZ FLÓREZ, Luis Carlos. Planeación de Proyectos Informáticos. Grupo de Investigación en Sistemas y Tecnología de la Información STI. Bucaramanga, 2001

JAMSA, Kris. Superutilidades Para Visual Basic.NET. McGraw-Hill, 2003

SHAPIRO, Jeffrey R. Visual Basic.NET : Manual De Referencia

http://www.det.uvigo.es/~jose/doctorado/proceso_sw/sld013.htm

<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node11.html>

<http://www.microsoft.com/spanish/MSDN/estudiantes/ingsoft/ingenieria/default.asp>

<http://www.elguille.info/NET/cursoVB.NET/indice.htm>

<http://www.microsoft.com/spanish/msdn/vs2005/default.mspx>

<http://www.microsoft.com/latam/sql/>